Expressive timing analysis in classical piano performance by mathematical model selection.

Li, Shengchen

Expressive timing analysis in classical piano performance by mathematical model selection

Shengchen Li

Submitted in partial fulfillment of the requirements of
the Degree of Doctor of Philosophy

16th March 2016

Abstract

Given a piece of music, the timing of each beat varies from performer to performer. The study of these small differences is known as expressive timing analysis. Research into expressive timing helps us to understand human perception of music and the production of enjoyable music. Classical piano music is one music style where it is possible to measure expressive timing and hence provides a promising candidate for expressive timing analysis.

Various techniques have been used for expressive timing analysis, such as the Self-Organising Map (SOM), parabolic regression and Bayesian models. However, there has been little investigation into whether these methods are in fact suitable for expressive timing analysis and how the parameters in these methods should be selected. For example, there is a lack of formal demonstration that whether the expressive timing within a phrase can be clustered and how many clusters are there for expressive timing in performed music. In this thesis, we use a model selection approach to demonstrate that clustering analysis, hierarchical structure analysis and temporal analysis are suitable for expressive timing analysis.

Firstly in this thesis, we will introduce some common methods for model selection such as Akaike's Information Criterion, Bayesian Information Criterion and cross-validation. Next we use these methods to demonstrate the best model for clustering expressive timing in piano performances. We propose a number of pre-processing methods and Gaussian Mixture Models with different settings for covariance matrices. The candidate models are compared with three pieces of music, including Balakirev's Islamey and two Chopin Mazurkas. The results of our model comparison recommend particular models for clustering expressive timing from the candidate models.

Hierarchical analysis, or multi-layer analysis, is a popular concept in expressive timing analysis. To compare different hierarchical structures for expressive timing analysis, we propose a new model that suggests music structure

boundaries according to expressive timing information and hierarchical structure analysis. We propose a set of hierarchical structures and we compare the resulting models by showing the probability of observing the boundaries of music structure and showing the similarity of the same-performer renderings. Our analysis supports the proposition that hierarchical structure improves the performance of modelling over non-hierarchical models for the performances that we considered.

Researchers have also suggested that expressive timing is influenced by music structure and temporal features. In order to investigate this, we consider four Bayesian graphical models that model dependencies between a position in a music score and the expressive timing changes in the previous phrase, on expressive timing in the current phrase. Using our model selection criterion, we find that the position of a phrase in music scores is only shown to effect expressive timing in the current phrase when the previous phrase is also considered.

The results in this thesis indicate that model selection is useful in the analysis of expressive timing. The model selection methods we use here could potentially be applied to a wide range of applications, such as predicting human perception of expressive timing in music, providing expressive timing information for music synthesis and performance identification.

Contents

Statement of Originality

I, Shengchen Li, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author.

Signature: SHENGCHEN LI

Date: 16th March 2016

Acknowledgement

First of all, I would like to give my special thanks to Chinese Scholarship Council (CSC) that funded me for past four years.

Secondly, I would like to thank Queen Mary University of London, where I spent my four most glossary year in my life and where I recognise my precious friends.

Next, I would like to say thank you to my parents who stand behind all the times, support me in various means and are always proud of their son.

Also, I would like to thank my piano teacher, Mrs. Jin Ju, who taught me to play the piano almost twenty years ago. The experience of playing the piano inspires me for the idea of writing this thesis.

Finally, I would like to show the my sincere appreciation to who are or were members of my supervision team: Dr. Dawn Black, Prof. Elaine Chew, Prof. Mark Plumbley and Dr. Simon Dixon. Only with your contributions can I produce such a thesis in my early research career.

In summary, I would like to use this thesis to memorise my most glorious time of my life in London.

This thesis was proof-read for the purposes of spelling and grammar by Wordy^{TM}.

Chapter 1

Introduction

1.1 Motivations

In performed music, performers impart their understanding of a piece of music by varying different aspects of their performance, such as tempo and dynamics. It is interesting to investigate how performers vary these aspects of a performance to give expression to a piece of music. Past works analysing expressiveness cover various topics such as timing [Repp, 1998], dynamics [Repp, 1999a], finger motion [Goebl and Palmer, 2009], and piano pedal [Chew and François, 2008].

The timing of the beats in a piece of performance, or beat timing, is a major topic within the analysis of expressiveness. In most cases, beat timing exhibits a close link with tempo. Tempo can be defined as "the rate at which musical notes are played, expressed in score time units per real time unit" [Dixon, 2001]. However, research into tempo relates to the perception of tempo. For example, Cambouropoulos et. al. [Cambouropoulos et al., 2001] explored perceived tempo preferences, concluding that humans prefer a smoother version of the original tempo. This thesis focuses on the variation of beat timing. For clarity, the term "expressive timing" represents the variation in the timing of each beat. In our experiments, tempo is used to measure expressive timing, or, say, the length of each beat. This general principle holds throughout this thesis.

It would not be sensible to investigate the expressive timing for all pieces of music as there are so many of them. As a result, we need a certain number of candidate performances to form a database for investigation. These are usually classical music as it has more variations in beat timing than any other type of music. The richness of variations in the expressive timing makes its analysis easier. Moreover, piano music is preferred by most researchers, because keyboard instruments are particularly good at obtaining a precise tempo as the attack of the keyboard is more easily defined.

Investigating the variations in expressive timing has interested many researchers such as Repp [Repp, 1993], Sprio et al. [Spiro et al., 2010] and Widmer et al. [Widmer et al., 2010]. There are certain common methods applied to researching expressive timing regardless of the focuses. For example, 1) clustering tempo variations within a certain part of performances for further analysis [Spiro et al., 2010]; 2) making use of positional and temporal features within a piece of performance to synthesise possible expressive timing ([Todd, 1992];[Widmer et al., 2010]); and 3) considering the hierarchical relationship to synthesise expressive timing or to model the tempo variation ([Todd, 1992]; [Widmer and Tobudic, 2003]).

Although these methods have been applied in a wide range of research and have been demonstrated useful by the results of experiments, in some cases, there lacks a formal justification of some experiment variables in the experiment such as the number of clusters chosen for tempo variations within an excerpt music. Within the field of machine learning and mathematics, there is a methodology called the model selection test [Claeskens and Hjort, 2008, p. 1]. In a model selection test, different indicators are proposed to evaluate the performance of candidate models. In this thesis, we perform a series of model selection tests to support the selected methods used for analysing expressive timing. In each model selection test, we propose a number of models under different hypothesis. We then compare the model performance to select the candidate model whose hypotheses is best supported by the model selection test.

There are several common model selection tests used including model selection criterion and cross-validation tests. In this thesis, we introduce these methods of model selection to the analysis of expressive timing in performed music. We propose several experiments that make use of model selection tests in order to verify some methods used in past works.

1.2 Research Goals

There are two main contributions in this thesis. Firstly, we demonstrate that the selected methods for analysis of expressive timing are supported by mathematical selection tests. Secondly, we draw conclusions based on the results of our model selection tests. With these conclusions, we demonstrate that the model selection tests can be enlightening when analysing expressive timing in performances.

The methods for analysing of expressive timing that we verify in this thesis are: clustering of expressive timing; using positional features to predict expressive timing of a phrase and analysing expressive timing hierarchically. We test these methods using three series of model selection tests. The proposed models are intended to verify the following hypotheses: 1) clustering is a suitable method for analysis; 2) the expressive timing of a current phrase is impacted by its position as well as the expressive timing of the previous phrase; and 3) analysis of expressive timing should consider hierarchical relationships.

The first topic we discuss in this thesis is clustering of expressive timing. In different parts of a performance, the general gestures of expressive timing are limited. In some relevant literature such as that by Rink et al. [Rink et al., 2011], Madsen and Widmer [Madsen and Widmer, 2006], clustering tempo variations within a certain unit forms a basis for further analysis. However, there appears to be no evidence showing that clustering is a suitable way of analysing expressive timing within a phrase. Furthermore, the reasons behind the choice of the number of clusters are not fully explained in most literature. In the majority of cases, this is because they are determined empirically.

We build up several non-clustered and clustered models to compare their performance when predicting the distribution of expressive timing within a phrase. We expect the model selection test to show that those clustered models with a certain number of clusters outperforms the other models. As a result, we can assert that the expressive timing within a phrase should be clustered.

Next we investigate which factor has the most impact on the strategy of expressive timing for a particular phrase. Building on the past works of Todd [Todd, 1992] and Widmer et al.[Widmer et al., 2010], we investigate two candidate factors: the position within the music score and the changes of expressive timing in the previous phrase. We consider the two factors not only individually but also in combination. We design a model selection test that draws on several Bayesian graphical models. We also propose a novel model selection criterion that balances the model complexity and performance. We then compare the candidate models and select the best performing model in terms of impact on expressive timing for a particular phrase.

The final experiment demonstrates that considering hierarchical structure is helpful for analysing expressive timing. The hierarchical analysis of expressive timing has been used several times in past research such as that by Sapp [Sapp, 2008], Tobudic and Widmer [Tobudic and Widmer, 2003b] and Todd [Todd, 1992]. To compare analysis of expressive timing with different structures, including different hierarchical structures and non-hierarchical structures, we propose a model that converts expressive timing to a probability for every beat in the performance that locates a boundary of music structure. We evaluate the resulting models that consider different structures by calculating the probability that the boundaries of the music structure can be observed and by showing the similarities between same-performer renderings.

In these three experiments, we use model selection tests to verify the use of common methods for analysing expressive timing. We also show that the results of these experiments can provide some statistical principles for expressive timing in performed music. Although our focus is the introduction of model selection tests for the analysing expressive timing, the demonstration of such principles

is valuable for analysing performed music.

1.3 Thesis Structure

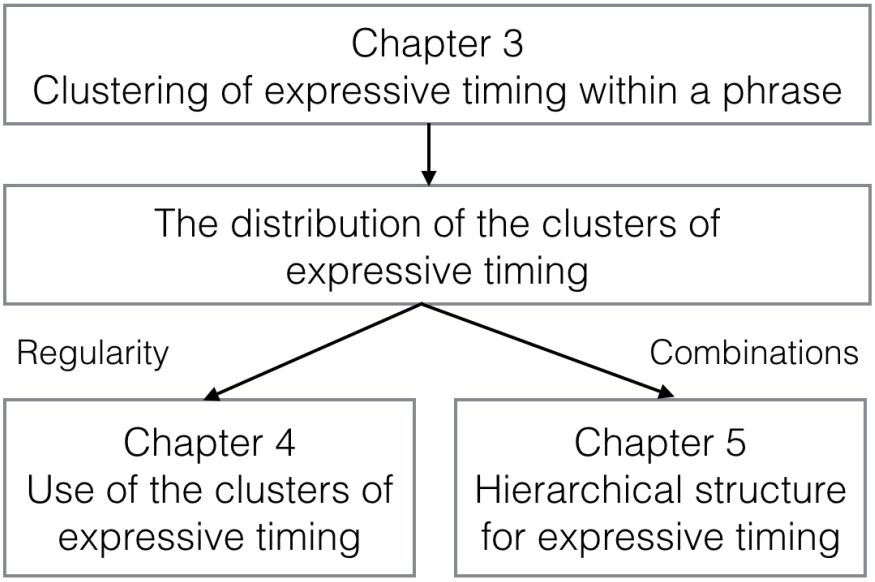


Figure 1.1: The relationship between the three topics in this thesis

Although the topics we have chosen in this thesis have been discussed in past works, these topics are related to each other as demonstrated by Figure 1.1. The first topic is the analysis of expressive timing within a phrase. We demonstrate that the expressive timing within a phrase can be clustered. The next step is to analyse the distribution of the clusters. This throws up a certain number of problems for further investigations in terms of: 1) the distribution of clusters; and 2) the impacts of phrase length.

Examining the sequential and positional distribution of clusters can be interpreted as investigating which factor impacts the use of clusters for performers. Furthermore, investigating the sequential links between two neighbouring phrases is equivalent to analysing the distribution of the combination of clusters. This effectively involve analysing longer phrases in a piece of performance.

If we combine a different number of clusters to form longer phrases, we are introducing a hierarchical structure for analysing expressive timing.

In summary, verifying the premise that expressive timing can be clustered forms a basis for understanding the relationship amongst the three topics in this thesis. The other two topics — investigating the factors that affect deciding on the clusters of expressive timing and investigating the hierarchical relationship between different phrase lengths — can be thought of as examining the distribution of clusters of expressive timing from different points of view.

1.4 Contributions

In this thesis, we

- introduce model selection to analyse expressive timing in computational musicology;
- demonstrate that the expressive timing within a phrase can be usefully clustered;
- show whether the position of a phrase and the expressive timing in the previous phrase impacts expressive timing in a phrase; and
- investigate whether considering the hierarchical structure is valid when analysing expressive timing.

1.5 Related Publications

There are two publications in proceedings of conferences related to this thesis including [Li et al., 2014] and [Li et al., 2015]. Both publications address the clustering of expressive timing within a phrase (Chapter 3). We go further than [Li et al., 2015], as we also show some evidence of the relationship between the other topics discussed here. The authors of these publications include all supervisors who contributed to the supervision of the works. The authors of [Li et al., 2014] also include Professor Elaine Chew, who contributed the music perception perspective.

This thesis is organised in the following way: first we give a literature review of related works in Chapter 2. Then we show the process of model selection tests in Chapter 3, Chapter 4 and Chapter 5. Each of these three chapters presents one model selection test that demonstrates a particular method for analysing expressive timing. Finally, we conclude the thesis with a summary and some ideas for possible future work.

Chapter 2

Background

In this chapter, we first review some works on the subject of computational musicology. Whilst this is not a computational musicology thesis, a comparison with that field allows us to demonstrate that our model selection tests contribute to the analysis of expressive timing.

Next we review related works concerning common methods of analysing expressive timing that we are going to address in this thesis. By reviewing these methods, we define the problem we are going to solve and discuss how model selection tests contribute to the analysis of expressive timing. Finally, we review common model selection tests. Introducing model selection tests to the analysis of expressive timing is one of the main contributions of this thesis.

2.1 Expressiveness and Musicology

Before reviewing specific past works, we would like to review some general topics related to this thesis. The term *expressive timing* in the title is related to the changes of timing about expressiveness. Snyder [Snyder, 2000] discussed expressiveness, stating

> The patterns of rhythm, melodies, and so on that we are able to remember from music consist of sequences of musical categories. Each occurrence of a category, however, is shaded in a particular way by

its nuance, which constitutes the expressive aspects of the music.

The term *expressive nuances* are 'continuous variations in the pitch or rhythm of a musical event'.

The expressive nuances are "continuous variations in the pitch or rhythm of a musical event". Many parameters are considered to contribute to expressiveness. The most-studied expressive features are tempo [Widmer, 2003, Repp, 1998, Grosche et al., 2010, Widmer and Tobudic, 2003] and dynamics [Repp, 1999a, Grosche et al., 2010, Widmer and Tobudic, 2003]. For piano music, pedal usage [Chew and François, 2008, Repp, 1996] and piano key touch [Kinoshita and Furuya, 2007, Goebl et al., 2004a, Goebl and Palmer, 2009] have also been studied.

Besides considering each factor individually, some works, such as [Repp, 1999b], also considered multiple perspectives of expressiveness. One of the most famous systems considering the multiple dimensions of expressiveness is the KTH system [Friberg et al., 2006], which is named after the research institute at which it was developed. This system considers multiple factors affecting expressiveness, including phrasing, micro-level timing, metrical patterns, grooves, articulation, tonal tension, intonation, ensemble timing and performance noise. In addition to the above factors, some factors beyond music performance, such as emotion [Livingstone et al., 2007], have also been considered in analyses of performed music.

Moreover, the factors listed also affect each other. For example, tempo and dynamics are two closely related aspects in performed music. Several investigations into tempo and dynamics have been undertaken, such as the studies on the Performance Worm [Dixon et al., 2002]. In addition to dynamics, Repp [Repp, 1996] asserted that pedal timing in piano performances may interact with expressive timing and melody. Beran and Mazzola [Beran and Mazzola, 2000] attempted to investigate the relationship between melodic, harmonic, metrical features and expressive timing. They concluded that the relationship between melodic, harmonic, metrical features and expressive timing is very complicated. Sundberg et al. [Sundberg et al., 2003] tried to use the rules in the KTH system

inversely to recover expressive timing in performed piano music. They found that structural information of the music could be an important factor in experiments. The research considering multiple dimensions of expressiveness suggests that expressive timing does not affect expressiveness of performed music as an independent factor.

Studies of expressiveness are commonly used to solve musicology problems. For example, empirical data in expressive performances have been used in studies of expressive style [Spiro et al., 2010], especially historic style changes [Timmers, 2007, Leech-Wilkinson, 2010]. Goebl et al. also used expressive data in performed music to investigate the between-hand synchronisation problem [Goebl et al., 2009] and the personal style-changing problem [Grachten et al., 2009].

2.2 Computational Musicology

In [Coutinho et al., 2005], computational musicology is broadly defined as 'the study of music by means of computer modelling and simulation'. Computational musicology involves a process that uses statistical models to represent expressive parameter variations for multiple purposes. Kirke and Miranda [Kirke and Miranda, 2013] introduced a general framework for computational modelling of expressive performances (Figure 2.1). The raw material commonly used for analysis and modelling is annotation from performances. In most cases, the database is collected from human performances. Music features such as melody and musical structure may also be extracted from the original performances to help the analysis. The performance context refers to the features of expressiveness in neighbouring positions within the performances. For example, Todd ([Todd, 1992]) asserts that tempo variations in expressive performances have a multi-layer structure and share a parabolic shape. The adaptation process extracts rules and models from performance data. Such rules and models are known as performance knowledge, which describes how performers control expressive actions. The terms "Instrument model" and "sound" in Figure 2.1

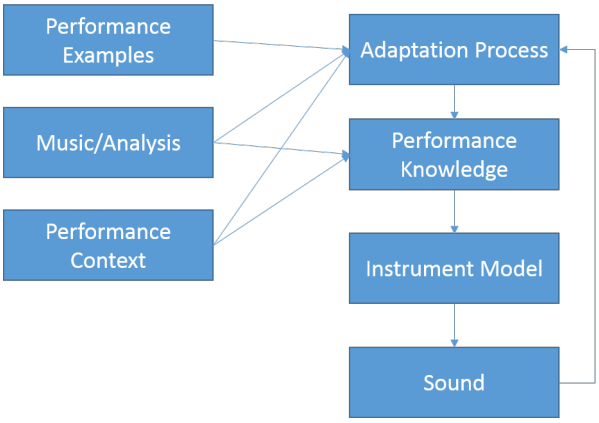


Figure 2.1: Generic framework for computational modelling of expressive performances [Kirke and Miranda, 2013]

relate to music synthesis, which is not the focuses of analysing expressive timing.

In summary, computational musicology is based on the remaining five parts shown in Figure 2.1. The new methodology of model selection we are introducing to computational musicology in this thesis, model selection tests, corresponds to the adaptation process shown in Figure 2.1. Next we are going to review a selection of representative works in computation musicology.

Widmer and Goebl [Widmer and Goebl, 2004] and Kirke and Miranda [Kirke and Miranda, 2013] have written good overviews of computational models of expressive music. In [Widmer and Goebl, 2004], Widmer has categorised computational models of expressive music performance into four categories: the rule-based model (such as KTH model, named by the Royal Institute of Technology in Sweden), Todd's model, mathematical models and machine learning models. A few years later, in [Kirke and Miranda, 2013], Kirke and Miranda classified computer systems of expressive music performance into non-learning models, linear regression, artificial neural networks, case and instance-based systems, statistical graphical models, evolutionary computation and other regression methods. Comparing the classifications in [Widmer and Goebl, 2004] and [Kirke and Miranda, 2013], we can see that machine learning and pattern recognition technologies have been widely used in recent years. Further, Kirke and

Miranda [Kirke and Miranda, 2013] suggested classification according to purpose of computational musicology models. In general, there are two purposes of expressive music performance computer systems: "analyse-for-synthesis" and "measurement-for-analysis". These purposes introduce another way of classifying computational music models. An "analyse-for-synthesis" system aims to synthesise expressive performance. A "measurement-for-analysis" system aims at understanding expressive actions in human performances. As a result, the output of such systems should show a clear understanding of expressive actions.

Here, we give a quick review of selected computational models of expressive performance. Traditional rule-based methods extract rules from musicology research. Existing musicology theory and even interviews of performers [Johnson, 1991] may provide source of such rules. In recent decades, machine learning algorithms have extracted the regularity in performance data. Next we are going to review a certain number of previous works on the subject of computational musicology.

The traditional rule-based system assumes that performers follow a set of fixed rules when they play [Widmer and Goebl, 2004]. Some rule-based systems, such as the Bach fugue system [Johnson, 1991], generate expressive performances with the same set of rules all the time; thus, they are less flexible. The KTH system [Friberg et al., 2006] proposed a parameter (k) to make the effects of the rules adjustable.

Todd's model [Todd, 1992] is another computational musicology work. Todd proposed a hierarchical parabolic model that regresses tempo curves using parabolic functions over musical structure. Todd concluded that tempo curves can be approximated by summing up multi-level parabolic curves. Todd's conclusion has been widely accepted by other research, especially for short passages of performance. The hierarchical approach is intended to exploit hierarchical music structure. Unlike the KTH system, Todd's model can be used to characterise the type of expressiveness portrayed in a piece of performed music.

Mazzola and Zahorka [Mazzola and Zahorka, 1994] use a mathematical method to analyse music expressivity. They represent musical material as a

multi-dimensional space mapped by onset time, pitch and duration. The variations in performed music (such as tempo variations) can be considered the result of mathematical transformation from the space of the musical material. Such mathematical transformation can occur through a generalised rule of expressiveness such as the rules in the KTH system. This research combines mathematics with computational musicology research and can be considered as evidence supporting the use of mathematical model selection methods for expressive timing analysis.

Besides the above non-learning musicology research, machine learning has been widely used in computational musicology in recent decades as an analysing technique. Typically, machine learning is used for two types of task: 1) finding expressive rules or model parameters from performance data (such as [Widmer and Tobudic, 2003]) and 2) recognising expressive patterns within expressive parameters from performance data (such as [Widmer et al., 2010]). Tobudic and Widmer introduced the DISTALL system in [Widmer and Tobudic, 2003], [Tobudic and Widmer, 2003a], [Tobudic and Widmer, 2003b]. Performances are decomposed into different levels of hierarchical groupings and analysed by an enhanced parabolic model, which considers the resilience of regression. To synthesise expressive performances, different levels of groupings are compared and weighted. Widmer, Flossmann and Grachten [Widmer et al., 2010] use the YQX (a simple Bayesian model) algorithm to synthesise expressive performances. The system first learns how performers vary expressive parameters at different places with the music score. Given a new musical score, YQX then identifies similar parts within the new score and synthesises performances by applying the patterns from the learned performances to similar parts of the new score. This method provides a reasonable quality of expressive performance. A recent research development introduced an interesting way of synthesising expressive performance: evolutionary computation [Miranda et al., 2012]. Multiple agents generate an expressive performance according to a set of rules and parameters. At the same time, each agent is listening to other agents and modifying their own parameters if

they evaluate the other agents' performances as being better. These approaches demonstrate that it is possible to use machine learning methods to build up models for expressive timing.

In machine learning, there exists a method called model selection [Burnham and Anderson, 2002, p. 13] that can be used to select the best mathematical model for a set of data. If there are several candidate models making different assumptions, we can select the best hypothesis by selecting the best model. In this thesis, we demonstrate that model selection methods can be applied to computational musicology.

2.3 Tempo Variations in Expressive Performances

In this thesis, we investigate the expressive timing of beats in performed music. Variations in beat timing can be measured using the lengths of the beats. Tempo is a more commonly used concept that can be defined as the reciprocal of beat length. Dixon [Dixon, 2001] defined tempo as: "the rate at which musical notes are played, expressed in score time units per real time unit".

Suppose the timing of each beat in a performance is $\{t_1, t_2, \ldots, t_n, t_{n+1}\}$, where t_{n+1} represents the end of the last note in the performance analysed. The tempo is defined as the rate of beat on each beat. The tempo during a particular beat can thus be calculated as the reciprocal of the beat duration, namely,

$$\tau_i = \frac{1}{t_{i+1} - t_i}. \tag{2.1}$$

In this thesis, analysis of expressive timing is consdiered to be analysis of tempo variations. However, the tempo can also be a perceptual concept; thus, we use the term "expressive timing" instead of "tempo variations" in most cases to avoid possible misunderstanding.

Now we will explore the methods for extracting original data from performances. Extracting the tempo values of each beat is known as beat tracking or beat detection. There have been many advances in automatic or semi-automatic

beat detection. Dixon and Gouyon gave a detailed review of related music information retrieval works in [Gouyon and Dixon, 2005]. In general beat tracking has two tasks: 1) analysing the spectrum to identify candidate pulses; and 2) determining the actual beat position amongst candidates' pulses. For expressive performances, selection of the correct candidate pulse is a tough task that often introduces false beats. For example, BeatRoot [Dixon, 2006] is a reasonable algorithm for beat tracking, but there remains a considerable amount of progress that can be made in terms of accuracy. Another automatic annotation method that can avoid false beats is MATCH [Dixon and Widmer, 2005]. The MATCH algorithm can align two performances of the same piece of music. In this research, a MIDI score is aligned with real performance. Thus, by marking the beat points in the MIDI file, the MATCH algorithm can annotate the aligned performance with beat points. MATCH can then avoid false or missing beats. This is a significant improvement over other beat tracking algorithms. Grosche et al. [Grosche et al., 2010] summarise which types of musical event affect the accuracy of beat tracking. This work summarised the problem of beat tracking when used for annotating beat timing in performed music. A more recent study of beat tracking is that by Fillon et al. [Fillon et al., 2015], which claims the statistical accuracy of beat tracking is under 0.9. Such beat tracking accuracy is still not suitable for annotating beat timing in performed music with a massive database. As a result, our method of annotating beat timing in this thesis still involves human input.

Next we review the common methods we selected for analysing expressive timing. These methods include: 1) clustering of expressive timing within a phrase; 2) the impacts of the position of a current phrase and the expressive timing in the previous phrase on the expressive timing in a phrase and 3) the use of hierarchical structure for analysing expressive timing.

Before getting into any detailed discussion, we would like to introduce the methods used for visualising the variations of tempo. A common way of visualising tempo variation is using a tempo curve [Repp, 1995a] that connects a series of tempo values on each beat within a specific part of a performance.

Figure 2.2 presents two different tempo curves across the first 84 bars of *Islamey* [Balakirev, 1902] in two performances (labelled as "P1" and "P2"). From these tempo curves, we can see that the tempo variations throughout this part of the performance is not totally free and that tempo variations in different performances share some similarities.

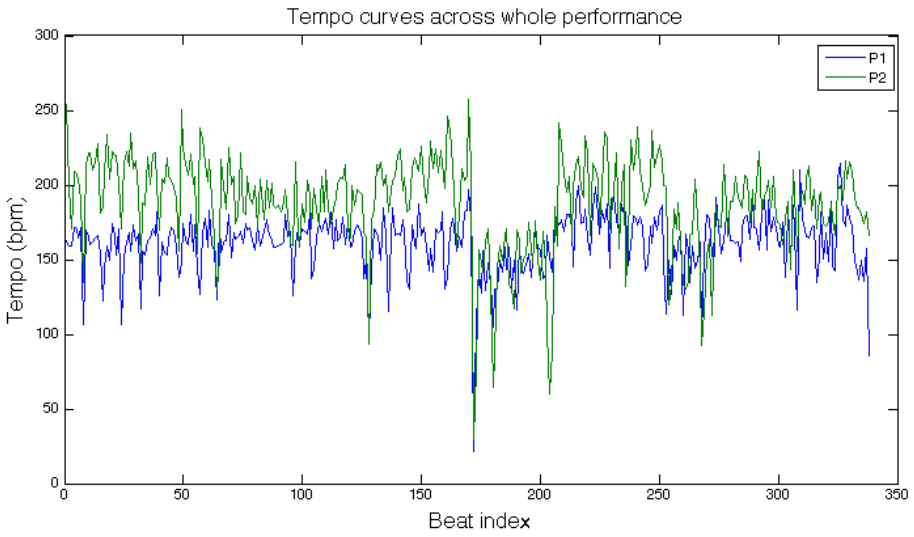


Figure 2.2: Two tempo curves across the first 84 bars of *Islamey*

In this thesis, we classify tempo variations into two categories: "intra-phrase" and "inter-phrase". The "intra-phrase" variation is the tempo changes on each beat within a phrase. The "inter-phrase" variation describes how the expressive timing within a phrase differs from that in other phrases.

According to [Nattiez, 1990], a music phrase can be is defined as "a unit of musical meter that has a complete musical sense of its own".

In this thesis, to avoid possible arguments about identifying phrase boundaries, we informally define a phrase as "a unit of musical meter"; in other words, a part of music. Thus, we do not utilise the music sense in phrases as an input of our experiments. However, we believe that the phrases we have identified should also make musical sense as the identification of phrases in this thesis is either confirmed by a composer we engaged or provided by the database itself. In Figure 2.3, we present the first twelve phrases of the piece *Islamey* that we

have identified for analysis in this thesis. In this diagram, the alternative blocks represent the boundaries of phrases and the letter in each block represents the theme of phrase. The collection of the performances of *Islamey* is used as a private database in Chapter 3 and Chapter 4.

2.3.1 Intra-phrase tempo variations

In general, different methods are used for analysing expressive timing within a phrase, ranging from statistical analysis [Desain and Honing, 1994a], correlation [Sapp, 2007] to Principle Component Analysis (PCA) [Repp, 1998] and machine learning [Zanon and Widmer, 2003]. Desain and Honing [Desain and Honing, 1994a] used variance analysis to reveal relationships between tempo and timing. Sapp [Sapp, 2007] used correlation to measure the similarity of performances. Repp [Repp, 1998] uses PCA to compare the commonality of different performances. Zanon and Widmer [Zanon and Widmer, 2003] used a machine learning algorithm to recognise different performers by learning high-level features.

Another common approach used in previous studies is clustering tempo variations. Various research such as [Spiro et al., 2010], [Grachten et al., 2009], [Repp, 1995a] shows that intra-phrase tempo variations can be clustered. The techniques involved in clustering the tempo variations within a phrase include self-organising maps [Spiro et al., 2010] and hierarchical clustering [Grachten et al., 2009]. In Chapter 3, we are going to demonstrate that clustering is a suitable approach for analysing expressive timing by showing that a clustered model is better than a non-clustered model for predicting the unobserved data of expressive timing within a phrase.

2.3.2 Dependencies of expressive timing

In Figure 2.2, we notice that tempo curves from different performers are not totally different. At certain positions, performers reach an agreement of slowing down or speeding up. These facts suggest that tempo variations should not be considered as random or independent events in expressiveness. For example, De-

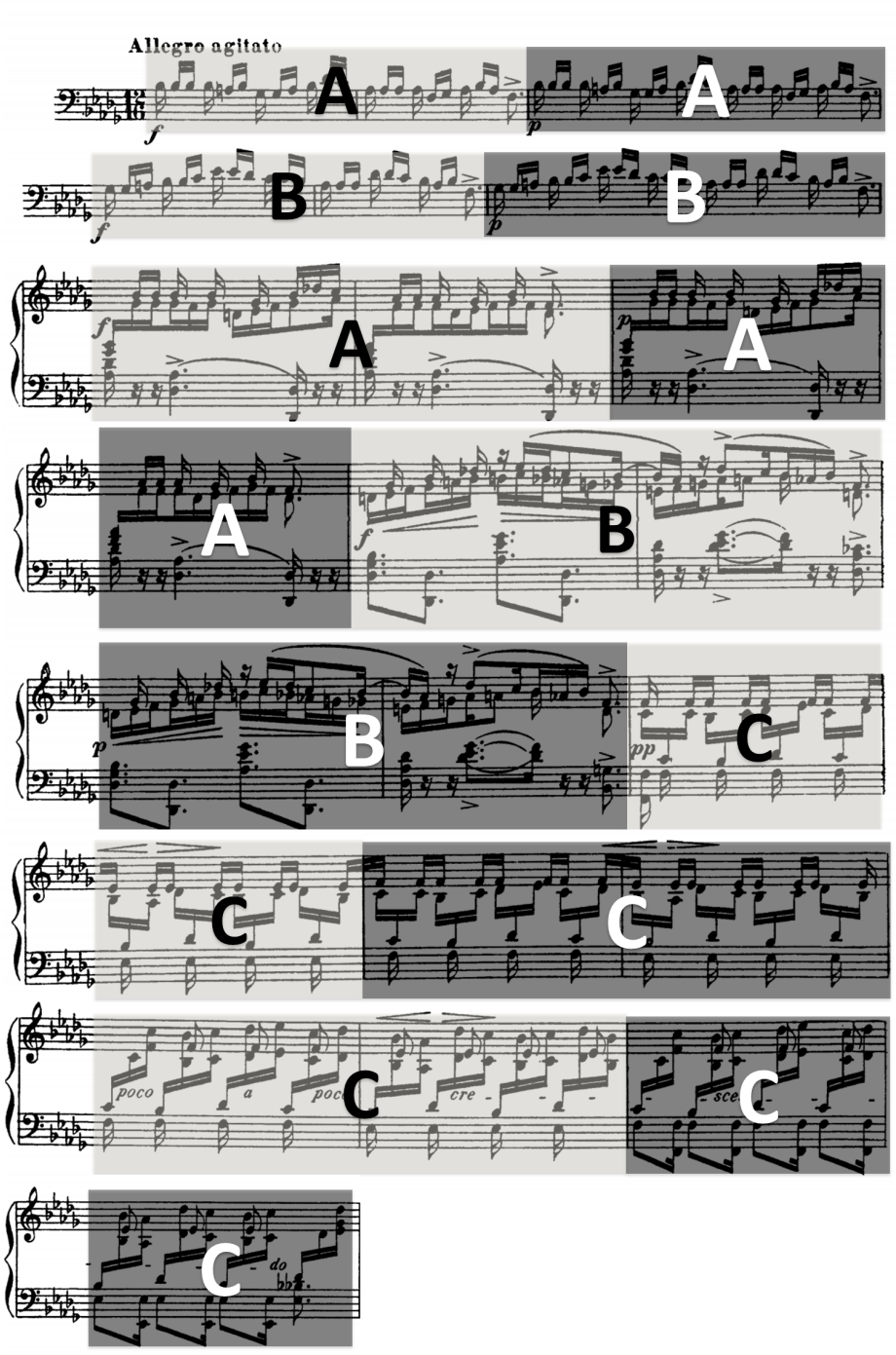


Figure 2.3: The first twelve identified phrases in *Islamey*

sain and Honing [Desain and Honing, 1993] asserted that tempo should not be considered as an independent contributor to expressivity but should be analysed with other expressive parameters such as loudness and musical structure. Goebl, Pampalk and Widmer [Goebl et al., 2004b] performed an analysis of loudness and tempo space and defined commonalities and differences in performances.

One particular research direction is to consider musical structure in conjunction with other performance data. Desain and Honing [Desain and Honing, 1993] scaled tempo curves across the whole performance at 60 beats per minute (bpm) to 90 bpm. The resulting performance sounded artificial. In later research, Desain and Honing [Desain and Honing, 1994a] suggested that tempo curves should be analysed along with consideration of musical structure. Repp [Repp, 1998] described detailed expressive principles by analysing a phrase of Chopin's Etude and asserted four factors of timing strategies involving melodic gestures. Bisesi et al. [Bisesi et al., 2011] modelled expressive timing near accents in performances defined as "local events that attract a listener's attention". Spiro, Gold and Rink [Spiro et al., 2010] analysed the beat timing inside of bars and classified bars into four clusters according to beat length distribution, asserting that musical structure and performed patterns are closely related. In Chapter 4, we investigate whether the structure of the music has any impact on the choice of the clusters of expressive timing for a particular phrase.

2.3.3 Hierarchical structures for expressive timing

In previous works, we find that the researchers considered expressive timing hierarchically. Having a hierarchical structure when analysing expressive timing is not rare. In the works we reviewed, a hierarchical structure is commonly assumed. For example, in [Todd, 1992], parabolic curves can also have a multi-level structure. This property is adapted by the DISTALL system ([Widmer and Tobudic, 2003]; [Tobudic and Widmer, 2003a] and [Tobudic and Widmer, 2003b]), which also considers the hierarchical relationship in tempo variations. However, the hierarchical relationship in tempo variations does not ensure the success of expressivity rendering. For example, Desain

and Honing [Desain and Honing, 1993] attempted to use the hierarchy relationship to synthesis a piece of MIDI, but had little success.

In Chapter 5, we investigate in two experiments whether a model that has a hierarchical structure outperforms one with a non-hierarchical structure. We propose a model that asserts the probability of every beat in a performance locating a boundary of the music structure according to expressive timing. We evaluate the performance of resulting models by calculating the query likelihood of the music structure boundaries of the piece. Another experiment is to show similarities between same-performer renderings. Formerly in Sapp [Sapp, 2008], Zanon and Widmer [Zanon and Widmer, 2003] attempted to recognise performers by measuring the similarities between performances by the same performer. We compare models of same-performer renderings and assess how well these models show the similarities between same-performer renderings.

2.4 Mathematical Model Training and Evaluation

Until now, we have been discussed the problems we are going to solve in this thesis. In this section, we are going to review the methods we use. We firstly introduce how we train our candidate models using a database, then we introduce how we assess the performance of the candidate models.

2.4.1 Model training

We review existing ways of training the candidate models in Chapters 3 and 4. In Chapter 3, we use a Gaussian models to represent non-clustered models as Gaussian model is the most widely used. The definition of a Gaussian model is:

$$\mathcal{N}(\tau|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(\tau-\mu)^2}. \tag{2.2}$$

A straightforward way of building a clustered model based on non-clustered models is mixing several non-clustered models, such as

$$p(\tau_i) = \sum_{a=1}^{A} \pi_a \mathcal{N}(\tau_i | \mu_a, \mathbf{\Sigma}_a). \tag{2.3}$$

This model is known as the Gaussian Mixture Model (GMM) with A Gaussian components.

In a GMM, there are three sets of parameters to be trained: the weight of each Gaussian component π_a, the mean of each Gaussian component μ_a and the covariance matrices $\mathbf{\Sigma}_a$. The most common method for training a GMM with data is the Expectation Maximum (EM) algorithm [Murphy, 2012, p. 350].

The EM algorithm ([Murphy, 2012, p. 350]) is an iterative algorithm and is commonly applied to mixture model learning. The training process of a given candidate model starts with a set of initial parameters. The performance of fitting a candidate model with this set of parameters is measured by model likelihood. This step is called the Expectation step (E-step). Next the model parameter is varied in order to maximise the model likelihood, which is called the Maximization step (M-step). This process continues until certain conditions are met. One of the most common conditions used is convergence; in other words, the model likelihood stops increasing. The final model is that with the highest model likelihood.

In Chapter 4, we introduce Bayesian graphical models for establishing different dependencies on the factors we propose. Bayesian parameter estimation ([Koller and Friedman, 2009, p. 741]) is a common method used for learning the parameters in candidate models. Unlike Maximum Likelihood Estimation (MLE, [Koller and Friedman, 2009, p. 722]), Bayesian estimation can solve the problem when some specific types of data that are not included in the training database but appear in the testing database.

2.4.2 Model evaluation

Once we have obtained our model with trained parameters, we assess the candidate models and investigate which model outperforms the others. There are

a few methods that measure the performance of candidate models such as Takeuchi's Information Criteria (TIC) ([Burnham and Anderson, 2002, p. 65]) and Minimum Description Length (MDL) ([Grunwald, 2005]). In this thesis, we use two methods. One is cross-validation [Burnham and Anderson, 2002, p. 36] and the other is model selection criterion [Burnham and Anderson, 2002, p. 37].

We can measure the performance of a model using model likelihood. The model likelihood represents the probability density that a specific set of observations comes from a specific model. If we present our dataset as $T = \{\overrightarrow{T_1}, \overrightarrow{T_2}, \ldots, \overrightarrow{T_n}\}$, then the model likelihood is $\mathcal{L} = p(\theta|T_n)$, where θ represents the parameter set of the candidate model $\mathcal{M}$. A model with better performance should have a higher likelihood [Murphy, 2012, p. 321].

A similar measurement of model performance for a model whose parameters are all discrete is query likelihood. This concept is used for evaluating language models [Manning et al., 2009, ch. 12]. Suppose there are N symbols in the dataset, then the probability mass distribution of these symbols can be represented as $\{p_1, p_2, \ldots, p_n\}$. If we have a model whose probability mass distribution of these symbols is $\{q_1, q_2, \ldots, q_n\}$ and there are M samples in the dataset, the query likelihood of the model for the dataset is

$$\mathcal{L} = \prod_{i=1}^{n} \prod_{j=1}^{n} q_i^{Np_j}. \tag{2.4}$$

For accuracy, we use logarithms to scale the query likelihood. Moreover, if we want to compare the query likelihood for different datasets, we compare the query likelihood for each sample on average. So the averaged logarithm form of query likelihood is

$$\frac{1}{N} \log \mathcal{L} = \sum_{i=1}^{n} \sum_{j=1}^{n} q_i \log p_j. \tag{2.5}$$

Unless otherwise specified, the query likelihood is represented in its averaged logarithm form. Moreover, in information theory, cross-entropy has the same form of (2.5) but with a different sign. In Chapter 4, we make use of this equiv-

alence between query likelihood and the cross-entropy to measure the model performance.

If we use only query likelihood or model likelihood to evaluate the performance of models, however, we cannot overcome a problem called overfitting [Murphy, 2012, p. 22]. If the model is very complicated such that it can fit the data in a dataset precisely but it fails to fit the unobserved data, the model is said to be overfitting the dataset. A good model selection method should avoid the problem of overfitting. Next, we are going to review the model selection methods that we use in this thesis.

The first way is model selection criterion. A model selection criterion balance the performance of models and the complexity of models. In this thesis we use two model selection criteria: the Akaike's Information Criterion (AIC) [Claeskens and Hjort, 2008, p. 22] and the Bayesian Information Criterion (BIC) [Claeskens and Hjort, 2008, p. 70]. Both model selection criteria penalise the model likelihood according to the complexity of the models. However, the penalty of model complexity is different for AIC and BIC. AIC penalises the model complexity without considering the number of samples in the dataset but BIC does consider the number of samples in the dataset when penalising the model complexity. If we use $\mathcal{L}$ to represent the model likelihood for a dataset containing N samples and we use $o(\theta)$ to represent the number of parameters in the model, the definition of AIC and BIC can be simplified as:

$$AIC = 2 * o(\theta) - 2 * \mathcal{L}, \tag{2.6}$$

$$BIC = log(N) * o(\theta) - 2 * \mathcal{L}. \tag{2.7}$$

In fact, AIC and BIC are two representative model selection criteria [Burnham and Anderson, 2002, p. 37]. The differences in penalising the model complexity give AIC and BIC different properties. AIC is an "efficient" model selection criterion [Claeskens and Hjort, 2008, p. 99] as AIC is good at choosing the best model that approximates the data in the training dataset. On the other hand, BIC is a "consistent" [Claeskens and Hjort, 2008, p. 107] model

selection criterion as BIC tends to select the model that predicts the distribution of unobserved data. For a detailed discussion, please refer to Chapter 4 in [Claeskens and Hjort, 2008].

In this chapter, we have reviewed some of the problems encountered in computational musicology. Although certain methods are used to solve these problems, there lacks formal demonstration to support the proposed methods in the previous works. This thesis demonstrates that such methods are usable for solving the problems mentioned in this chapter by using model selection methods to explore the clustering of intra-phrase expressive timing, the factors that impact the use of clusters and the hierarchical structure present in the analysis of expressive timing. We use these model selection tests to show that the methods proposed in the previous works can be supported by model selection tests and the model selection methods can be applied to computational musicology research. In the next three chapters, we are going to present how model selection methods are used to support the methods used in computational musicology research and how model selection methods are used for computational musicology research.

Chapter 3

Model Analysis for Expressive Timing within A Phrase

This thesis aims to show how model selection methods can be used for the expressive timing analysis of classical piano music. To start with, we investigate the expressive timing within a phrase. Although examining expressive timing within music is common in previous studies [Spiro et al., 2010], [Repp, 1995a], [Madsen and Widmer, 2006], the unit of length in each analysis varies from research to research, e.g. half a bar [Madsen and Widmer, 2006], bar [Spiro et al., 2010] and phrases [Repp, 1995a].

Many researchers in the previous studies often clustered expressive timing [Spiro et al., 2010], [Grachten et al., 2009], [Madsen and Widmer, 2006]; however, there is little evidence available to date to support the notion that expressive timing can be clustered. In this chapter, we demonstrate that expressive timing can in fact be clustered. The unit we chose for the expressive timing is the phrase, which we informally defined in section 2.3. We choose a phrase of music as a phrase can contain enough variations in expressive timing to enable us to perform an accurate analysis. Moreover, analysing the expressive timing

with the unit of a phrase can provide more samples than analysing with the unit of a performance with the same database. Also, using available samples can also potentially benefit the accuracy of machine learning.

To support the notion that expressive timing within a phrase can be clustered, we compare the performances of clustered and non-clustered models for fitting the distribution of expressive timing within a phrase. As there is no prior knowledge about how the expressive timing is distributed, we choose the Gaussian model — the most widely used non-clustered model [Murphy, 2012, p. 39] — as the framework of candidate non-clustered models. As a result, the mixture of Gaussian models — the Gaussian Mixture Model (GMM) — is chosen as the framework of the candidate clustered models.

Common methods used for comparing the candidate models, or model selection tests, include the use of model selection criterion, goodness-of-fit tests and cross-validation tests. We chose cross-validation as the primary measurement of model performance because "cross-validation has been suggested and well studied as a basis for model selection" [Burnham and Anderson, 2002, p. 36]. The use of model selection criteria is hence selected as our second evaluation of model performance for comparison purposes.

As well as determining the mathematical model and the methods of model selection, we also need a database for our analysis. For simplicity, we want the candidate performance to have identical lengths for each phrase. Furthermore, to aid clustering, we also want the candidate piece to be repetitive, as we anticipate the expressive timing in repetitive phrases to be similar to each other. The candidate piece we selected and utilised in this chapter is the first 84 bars of *Islamey* [Balakirev, 1902], which contains only three themes repetitively utilised. We also choose Chopin Mazurkas Op.24 No.2 (in short, Op.24/2) and Op.30 No.2 (in short, Op.30/2) to demonstrate the clustering of expressive timing within a phrase. With these three pieces of performance, we demonstrate that our conclusions can be potentially extended to other pieces.

This chapter is organised in the following way: we first introduce our performance database. Then, we introduce the clustered and non-clustered candidate

models. Next, we test the cross-validation likelihood of the candidate models and examine the relationship between the model selection criteria and the cross-validation likelihood with *Islamey*. Finally, we investigate whether similar results can be repeated for the two Chopin Mazurkas.

3.1 Data Collection

As we discussed, in this chapter, we use two databases: a public database and a private database.

The public database is the Mazurka dataset annotated by Sapp. The database is used as the raw data in [Sapp, 2008], [Spiro et al., 2008] and [Spiro et al., 2010] and was created by the CHARM project.[1] The Chopin Mazurkas have 3-beat bars and the music structure information is included in the database for each candidate piece. Mazurkas are popular pieces amongst classical pianists, and thus for each piece in the Mazurka database, there are multiple performances from the same performer. There are five pieces of Mazurkas in the database: Op.17/4, Op.24/2, Op.30/2, Op.63/3 and Op.68/3. However, as we discussed, we want the phrase lengths in the candidate pieces to be consistent, consequently we only used the data from Op.24/2 and Op.30/2 in this chapter.

The private database consists of 25 performances of *Islamey*. Unlike Mazurkas, which has a comprehensive but complicated music structure, the music structure of *Islamey* is simpler but the phrase lengths are consistent. The candidate piece in the private database is the first 84 bars of *Islamey* [Balakirev, 1902]. This section of *Islamey* has a four-bar coda and 40 two-bar phrases. In this database, we exclude the four-bar coda as the length of the coda differs from the other phrases, so in total we have 40 phrases for analysis in each performance. The initial structure analysis was performed personally and verified by a professional composer. Moreover, the composer we engaged pointed out that there are only three themes for the two-bar phrases in the part of *Islamey* we considered and that two themes repeat ten times and one theme repeats twenty times. We show

[1] www.charm.rhul.ac.uk

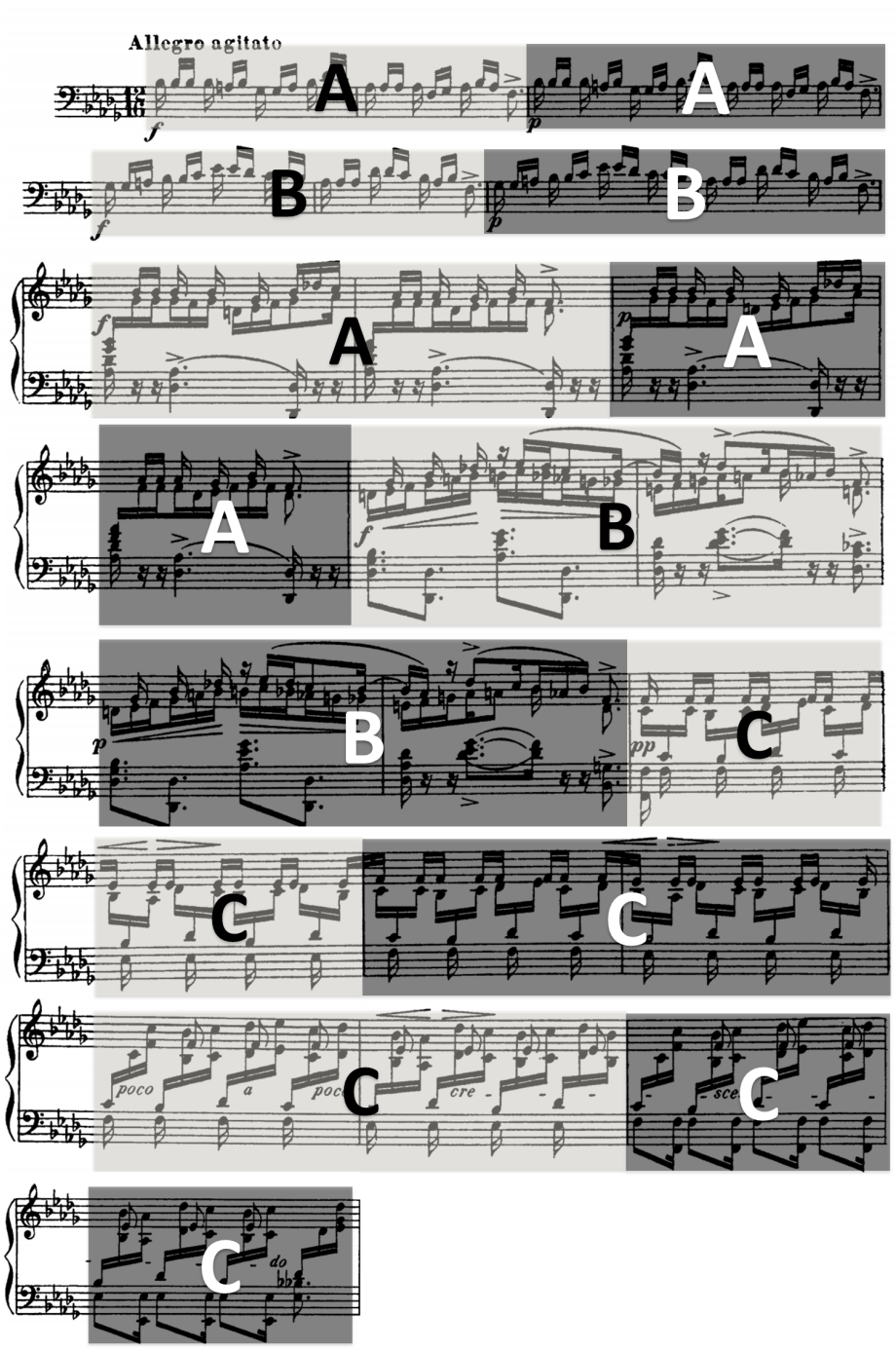


Figure 3.1: The music structure analysis of the first twelve phrases in *Islamey*.

the results of the analysis of the music structure in Figure 3.1, which is appeared in Chapter 2. We anticipate the expressive timing in repetitive phrases would be similar, thus the expressive timing in the *Islamey* database may lead itself to clustering. In our *Islamey* database, we have 25 performances from different performers (*See* **Appendix**). As there are 40 phrases considered in each piece of performance, in total we have $40 \times 25 = 1000$ annotated phrases in the *Islamey* database. Annotation of beat timing for all performances in the *Islamey* database takes about 75 hours. We give a detailed description of how we collect performance data of expressive timing in the *Islamey* database later in this section.

Both *Islamey* and the Chopin Mazurkas exhibit a hierarchical music structure. In defining the term phrase to specify the basic unit of music structure, we use the term, higher-level phrase, to mean a segment that contains several consecutive phrases. We also specify the length of phrases in the context of this thesis. In this chapter, the *Islamey* is used as the subject database due to its simplicity. The Mazurka database is included later to show that our methods are also useful with more complex music.

Now we are going to introduce how we annotated our expressive timing in *Islamey*. To minimise the error of annotation, we utilised a two-stage process for recording beat timing. This method makes use of the advantages of both human and machine annotation. First, we annotate beat timing by hand with keyboard tapping on a computer. Next, we adjust the annotated timing according to the values of the function for beat detection.

Currently, the accuracy of automatic beat detection is still lower than human annotation in performed music. As a result, the popular method of beat tracking is to tap along with the performed music [Grosche et al., 2010]. However, due to the perception process and possible delays from the devices [Degara et al., 2011], there are minor errors of beat timing in human annotation. To address this, we employ a beat tracking function (such as in [Davies and Plumbley, 2007]) for a more precise timing. In Figure 3.2, we show the two-stage method for the annotation of beat timing. The tool used for the annotation of beat timing is

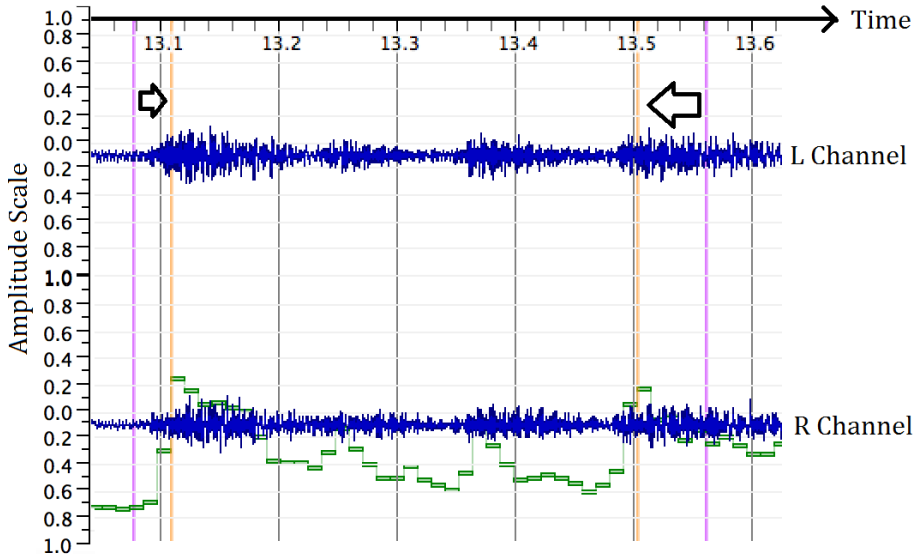


Figure 3.2: The annotation of beat timing

Sonic Visualiser[2]. The y-axis shows the amplitude scale of waveforms in the L and R channel of the original audio file. The x-axis shows the timing.

We first tap along with each performance ten times. Then, the timing of each beat is utilised as the averaged timing of the ten different taps, as shown as the orange line in Figure 3.2. We then use a beat detection function in Sonic Visualiser [Davies and Plumbley, 2007], which is shown as the green contour in Figure 3.2. The line is not smooth but rather it shows steps as the time span of each step is related to the width of the window in the algorithm. Then, we manually move the annotated beat timings to the nearest peak shown by the beat detection function. The arrows in Figure 3.2 show such moves and the orange lines label the final beat timing. The final beat timing could be decided by any timing within the time span of a step, but this decision does not affect the accuracy of our analysis as we later convert the beat timing to tempo by calculating the Inter Beat Interval (IBI) [Dixon, 2001] before our experiments.

Although expressive timing is the subject of this thesis, the term, tempo, is more commonly used by musicians. Tempo is defined as "the rate at which musical notes are played, expressed in score time units per real time unit"

[2]www.sonicvisualiser.org

[Dixon, 2001]. In this thesis, we calculate the value of tempo using IBI. Here, we let a series of expressive timings on each beat in a performance be represented as $\{t_0, t_1, t_2, \ldots, t_n\}$, the tempo value can then be calculated as:

$$\tau_i = \frac{1}{t_i - t_{i-1}} = \frac{1}{\text{IBI}}. \tag{3.1}$$

In common practice, the unit of beats per minute (bpm) for tempo is used, so the conversion between beat timing and tempo can be written as:

$$\tau_i = \frac{60}{t_i - t_{i-1}} = \frac{60}{\text{IBI}}. \tag{3.2}$$

3.2 Pre-processing

The exact timing of beats does not reflect the perception of tempo. As suggested by [Cambouropoulos et al., 2001], we smoothed the raw tempo by averaging the three neighbouring beats. Here, we suppose $\{\tau_1, \tau_2, ..., \tau_n\}$ represents the tempo values of each beat in a performance, the smoothed tempo values are then represented as $\{\bar{\tau}_1, \bar{\tau}_2, ..., \bar{\tau}_n\}$, where

$$\bar{\tau}_i = \frac{\tau_{i-1} + \tau_i + \tau_{i+1}}{3}. \tag{3.3}$$

Although all our tempo values are taken from the same piece (*Islamey*), different performers will play at a different overall tempo throughout different phrases, which is known as speed bias. This prevents the direct comparison of phrases and so the expressive timing in each phrase should be standardised.

In previous works ([Desain and Honing, 1994b] and [Repp, 1993]), a logarithm was used to standardise tempo variations. The standardisation process minimises the difference in global tempo across different performances. We therefore also try a logarithm (LOG) standardisation process. Moreover, in statistics, a standard way to normalise the differences between means in samples is to use standard scores [Spiegel and Stephens, 2011, p. 101], which standardise the mean and variance of data to a specific value. We propose this as a candidate standardisation method MVR (Mean-Variance Regulation). Additionally,

a previous work suggested that the tempo variations within a phrase are effected by the global tempo [Repp, 1995b]. Therefore, we consider two other methods that investigate if the tempo variations within a phrase are proportional to other hyper-parameters (such as the mean and range of tempo variations within a phrase). The first method we propose is Mean Regulation (MR), which forces the mean tempo value in each phrase to be 1. Another method we proposed is Range Regulation (RR), which forces the range of tempo in each phrase to a specific value.

We introduce the implementation of four standardisation methods: RR, MR, MVR and LOG. Here, we give mathematical definitions of these methods. Let $\overrightarrow{T} = (\tau_1, \tau_2, \ldots, \tau_n)$ and $\overrightarrow{T_s} = (\tau_1^{\text{stand}}, \tau_2^{\text{stand}}, \ldots, \tau_n^{\text{stand}})$ represent original and standardised tempo variation within a phrase, respectively, so we can give a mathematical representation of each standardisation method.

3.2.1 Range-Regulation (RR) standardisation

The range of tempo variation within each phrase is regulated to 1 in this standardisation method. Unlike the other standardisation methods, RR forces the variations to an absolute unified value. By unifying the range of tempo variations in each phrase, the differences in standardised global tempo between phrases are minimised. The RR standardisation can be represented as:

$$\tau_j^{\text{stand}} = \frac{\tau_j - min(\vec{T})}{max(\vec{T}) - min(\vec{T})} \text{ for } j = 1, 2, \ldots, n. \tag{3.4}$$

3.2.2 Mean-Regulation (MR) standardisation

This method forces the mean value of tempo variation within each phrase to 1, which ensures differences of global tempo between phrases are removed. The degree of stretching of tempo variations is set to the mean of each tempo curve. This method assumes that the degree of tempo variation is related to the global tempo and hence can be taken as a simpler version of the standard score that is used in statistics [Spiegel and Stephens, 2011, p. 101]. The MR standardisation

can be represented as:

$$\tau_j^{\text{stand}} = \frac{\tau_j}{mean(\vec{T})}, \text{ for } j = 1, 2, \ldots, n. \tag{3.5}$$

3.2.3 Mean-Variance-Regulation (MVR) standardisation

This method is a common method used in statistics. We force the tempo variation in each phrase to have a mean of 0 and a variance of 1. This methods is known as normalisation in signal processing and statistics. It is also called standard score in statistics [Spiegel and Stephens, 2011, p. 101]. The mathematical representation of MVR is:

$$\tau_j^{\text{stand}} = \frac{\tau_j - mean(\vec{T})}{std(\vec{T})} \text{ for } j = 1, 2, \ldots, n. \tag{3.6}$$

3.2.4 LOG-scaling (LOG) standardisation

This method log scale tempo variations within each phrase. As the logarithm suppresses both the speed bias and variance, we do not need to regulate the mean and variance of each tempo curve. The mathematical representation of LOG standardisation is:

$$\vec{T_s} = log_2(\vec{T}). \tag{3.7}$$

In Figure 3.3, we show some examples of standardisation. The standardisation methods employed from left to right are: none (original tempo variations), RR, MR, MVR and LOG. The four sample tempo variations represent four easily identifiable types of tempo variations within a phrase. If the tempo in a phrase keeps speeding up, we identify the tempo variation as 'accelerating'. If the tempo in a phrase speeds up and then slows down, we call the type of tempo variation a 'symmetric type' of tempo variations within a phrase. If the tempo in a phrase has varied across a minor range, we classify the tempo variation as 'constant'. Finally, if the tempo in a phrase slows down, we classify the tempo variations as 'decelerating'.

From Figure 3.3, we can see that the differences in the global tempo are eliminated by the MR and MVR methods only. The LOG and RR methods only reduce such differences. Moreover, the RR and MVR methods tend to even

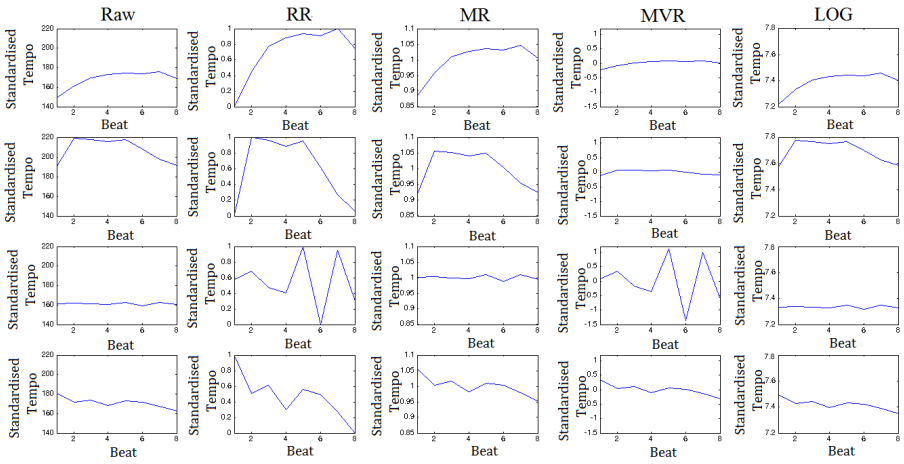


Figure 3.3: Comparison of the standardisation methods applied to different types of tempo variations (from top to bottom: accelerating type, symmetric type, constant type and decelerating type). The vertical axes in all the diagrams are the standardised tempo values (by definition of (3.4), (3.5) and (3.6), there are no units for the values of the standardised tempo. We omit the units for the standardised tempo of the LOG method for comparison purpose).

out the range of tempo variations across phrases. The MR method stretches the tempo variations in each phrase gently. Although the LOG method is a non-linear transformation, the shape of tempo variations changes very little, while the variations are slightly magnified. The MVR standardisation introduces variable results. As shown in the fourth column in Figure 3.3, the more variant tempo curves are flattened and the less variant tempo curves are amplified. However, as we are uncertain about which aspect affects the clustering of expressive timing, we also compared the experimental results with different standardisation methods employed in further experiments.

3.3 Mathematical Models

In our *Islamey* database, there are 25 performances and each performance comprises 40 phrases for analysis (*See section 3.1*). In each phrase, there are only

eight beats. As a result, the data we use for model analysis comprises 1000 samples of an eight-point vector. If we consider each eight-point vector as a point in eight-dimensional space, the candidate mathematical models predict the distribution of expressive timing in an eight-dimensional space. As we have no prior knowledge about the data of expressive timing, we use the most widely used distribution — a Gaussian distribution [Murphy, 2012, p. 39] — and its mixture to predict the distribution of expressive timing within a phrase. We used the Gaussian distribution as a non-clustered model and the GMM as a clustered model.

3.3.1 Non-clustered models

To build the Gaussian model, we need to train the mean and covariance matrix of the model. In this chapter, there are two different conditions for the mean and two different conditions for the covariance matrix. By combining the conditions for mean and the conditions for covariance, we obtained four candidate non-cluster models.

Besides the mean of the Gaussian model in the normal case [Murphy, 2012, p. 38], we propose a restriction on the mean as a series of constant values because in piano practice, using metronome to keep a constant tempo is considered a useful way to practise (in Prelude of [Franz, 1947]). As a result, if the mean is restricted, we only use the covariance matrix to fit the tempo variations within a phrase. We use the letter 'C' to represent the models with constant mean and the letter 'N' to represent the models that use the standard mean. Consequently, herein, the models with a constant mean are called 'C models' and the models with no restrictions on the mean are called 'N models'.

We propose two versions of the covariance matrix. The standard definition of the covariance matrix in Gaussian models has no restrictions. For comparison, we propose a restriction of the diagonal covariance matrix in other to investigate whether the tempo variation on each beat is related to the tempo variations on other beats. With the diagonal covariance matrix engaged, a multivariate Gaussian model can be written as the product of multiple Gaussian models,

which suggests that the variances of each beat are independent of each other. We use the letter 'F' to represent the standard definition of the covariance matrix and the letter 'D' to represent the models with a restricted covariance matrix. The restriction of the covariance matrix also has a musical significance as the restricted diagonal covariance matrix assumes the tempo variation on each beat is independent of tempo variations on other beats.

Combining the conditions for the mean and the covariance matrix in the Gaussian model gave us four types of non-clustered candidate models: CD models, CF models, ND models and NF models. Next, we give the mathematical definitions of the candidate models. However, before giving the definitions, we need to define some notations.

We use $\mathcal{N}$ to represent the Gaussian (Normal) distribution, $\overrightarrow{T_n}$ to represent the standardised tempo within a phrase, $\overrightarrow{\mu}$ to represent the mean of the Gaussian distribution and $\mathbf{\Sigma}$ to represent the covariance matrix. As we propose two types of means and covariance matrices, we use $\overrightarrow{\mu_c}$ and $\overrightarrow{\mu_n}$ to represent the means of the C and N models, receptively. Now if we let $\overrightarrow{T_i} = (\tau_{i1}, \tau_{i2}, ..., \tau_{ik})$ represent the standardised tempo variations in phrase i that has k beats, if there are l phrases in the database, then $\overrightarrow{\mu_c} = (\bar{\tau}, \bar{\tau}, ..., \bar{\tau})$, $\overrightarrow{\mu_n} = (\bar{\tau_1}, \bar{\tau_2}, ..., \bar{\tau_k})$, where $\bar{\tau} = \frac{1}{nl}\sum_{i=1}^{k}\sum_{j=1}^{l}\tau_{ij}$ and $\bar{\tau_i} = \frac{1}{l}\sum_{j=1}^{l}\tau_{ij}$. For the covariance matrix, we use $\mathbf{\Sigma}^{\text{full}}$ to represent the covariance matrix in the F model and $\mathbf{\Sigma}^{\text{diag}}$ to represent the covariance matrix in the D model. If we use σ_{kl}^2 to represent the covariance of beat k and beat l, thus σ_{kk}^2 represents the variance of beat k. We have

$$\mathbf{\Sigma}^{diag} = \begin{pmatrix} \sigma_{11}^2 & 0 & \dots & 0 \\ 0 & \sigma_{22}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_{nn}^2 \end{pmatrix} \text{ and } \mathbf{\Sigma}^{full} = \begin{pmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \dots & \sigma_{1n}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 & \dots & \sigma_{2n}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1}^2 & \sigma_{n2}^2 & \dots & \sigma_{nn}^2 \end{pmatrix}.$$

With the annotation introduced, we now define the four candidate models in (3.8), (3.9), (3.10) and (3.11), e.g.

$$p(\overrightarrow{T_n}) = \mathcal{N}(\overrightarrow{T_n}|\overrightarrow{\mu_c}, \Sigma^{diag}) \tag{3.8}$$

$$p(\overrightarrow{T_n}) = \mathcal{N}(\overrightarrow{T_n}|\overrightarrow{\mu_c}, \Sigma^{full}) \tag{3.9}$$

$$p(\vec{T_n}) = \mathcal{N}(\overrightarrow{T_n}|\overrightarrow{\mu_m}, \Sigma^{diag}) \tag{3.10}$$

$$p(\vec{T_n}) = \mathcal{N}(\overrightarrow{T_n}|\overrightarrow{\mu_m}, \Sigma^{full}). \tag{3.11}$$

3.3.2 Clustered models

A straightforward way to build a clustered model is to mix several non-clustered models [Murphy, 2012, p. 340]. Particularly in this research, we choose Gaussian Mixture Models (GMM) for comparison purposes because we select Gaussian models for non-clustered models. We recall that the definition of GMM with A Gaussian components for the distribution of multivariate variable τ_i is

$$p(\tau_i) = \sum_{a=1}^{A} \pi_a \mathcal{N}(\tau_i|\mu_a, \mathbf{\Sigma}_a). \tag{3.12}$$

There are three variables in GMM: the means of the Gaussian components μ_a, the covariance matrix of each Gaussian component $\mathbf{\Sigma}_a$ and the weight of each Gaussian component π_a. As in a GMM, as there is more than one Gaussian component, the means of all Gaussian components should be different. Moreover, since we have background knowledge about the weight of each Gaussian component, we cannot set restrictions on the weight. Therefore we propose some restrictions to the covariance matrices in GMMs only.

Similar to the case in non-clustered Gaussian models, we can restrict the covariances to be diagonal or not (namely to use $\mathbf{\Sigma}^{diag}$ and $\mathbf{\Sigma}^{full}$ in the proposed models, respectively). Again we use letter 'D' to represent the covariance matrices that are restricted to the diagonal and we use letter 'F' to represent the covariance matrices without restrictions. The musical significance of the restrictions of covariance matrices remains the same.

Furthermore, we want to investigate if the variance on each beat or the covariance between beats are independent to the tempo variations within a phrase. According to the definition of GMMs, there exists more than one covariance matrix in GMMs. We want to test if each covariance matrix is independent, thus we propose restricting the covariance matrix of each Gaussian component to be the same for comparison. We use letter 'S' to represent the restriction that the

covariance matrices of all the Gaussian components are the same and the letter 'I' to represent the normal GMM without restriction on the covariance matrices. Similar to the case of covariance matrices, we call the models with shared covariance matrices as S models and the models with independent covariance matrices as I models.

Combining the two types of restrictions we proposed for the covariance matrices in GMMs, we obtain four types of GMMs with various Gaussian components. If we use the letter $\mathcal{M}$ to represent the GMMs, the four types of GMMs are $\mathcal{M}_{SD}$, $\mathcal{M}_{SF}$, $\mathcal{M}_{ID}$ and $\mathcal{M}_{IF}$. We use a superscript to represent the number of Gaussian components, and the standardisation method used is included in brackets. For example, $\mathcal{M}^2_{SD}(RR)$ means a two-component GMM whose covariance matrix is diagonal and shared by Gaussian components, where the input data is standardised by RR. With the similar form of GMM definition in (3.12), the four candidate types of GMM are defined in (3.13), (3.14), (3.15) and (3.16) for the SD, SF, ID and IF models, respectively.

- GMM with shared diagonal covariance matrix $\mathcal{M}^A_{SD}$:

$$p(\tau_i) = \sum_{a=1}^{A} \pi_a \mathcal{N}(\tau_i|\mu_a, \mathbf{\Sigma}_a^{diag}), \text{ where } \Sigma_1^{diag} = \Sigma_2^{diag} = \ldots = \Sigma_n^{diag} = \Sigma^{diag}. \tag{3.13}$$

- GMM with shared full covariance matrix $\mathcal{M}^A_{SF}$:

$$p(\tau_i) = \sum_{a=1}^{A} \pi_a \mathcal{N}(\tau_i|\mu_a, \mathbf{\Sigma}_a^{full}), \text{ where } \Sigma_1^{full} = \Sigma_2^{full} = \ldots = \Sigma_n^{full} = \Sigma^{full}. \tag{3.14}$$

- GMM with independent diagonal covariance matrices $\mathcal{M}^A_{ID}$:

$$p(\tau_i) = \sum_{a=1}^{A} \pi_a \mathcal{N}(\tau_i|\mu_a, \mathbf{\Sigma}_a^{diag}). \tag{3.15}$$

- GMM with independent full covariance matrix $\mathcal{M}^A_{IF}$:

$$p(\tau_i) = \sum_{a=1}^{A} \pi_a \mathcal{N}(\tau_i|\mu_a, \mathbf{\Sigma}_a^{full}). \tag{3.16}$$

The term Σ^{diag} and Σ^{full} are defined in section 3.3.1.

The resulting GMMs can be used for the clustering of tempo variations. Each Gaussian component models a single cluster. A sample belongs to a cluster that has the maximum posterior probability for the respective Gaussian component. [Murphy, 2012, p. 342]

3.3.3 Remaining model parameters

To test the proposed models, two other parameters need to be determined. The first one is the standardisation method for tempo variations within a phrase. The other is the number of Gaussian components in the proposed models. We choose powers of 2 as possible numbers of Gaussian components (i.e. 2, 4, 8, 16, 32, 64, 128 and 256). We stop at 256 because the next possible number is 512 and the IF model would then have 47,736 parameters to be trained with 1000 samples, which has even more parameters than samples. Moreover, as the training process of GMMs is computationally expensive, training a 512-component GMM requires too much time, considering the computational power we have at our disposal.

The method we used for training a GMM is the Expectation Maximum method (EM method, for details see section 2.4.1). Since the initial parameter settings may lead to different resulting models in the EM algorithm, we repeat the training process of EM 1000 times for each type of GMM. In each training process, we start the training process with a different random initial value. Each resulting model is then evaluated by the model likelihood for the training dataset. The final result of each type of GMM is the model that has the highest model likelihood during the training process.

3.4 Model Evaluation Methods

The methods we used to evaluate the resulting model of each type of GMM are cross-validation and the use of model selection criteria. Cross-validation is known as "a basis of model selection". [Burnham and Anderson, 2002, p. 36]. However, cross-validation is a computationally expensive method, the use of

model selection criteria is sometimes used as an alternative method for model selection [Burnham and Anderson, 2002, p. 37]. In this chapter, we use both methods for model selection and examine how well they perform.

One of the commonly used variants of cross-validation is five-fold validation, where all data in divided into five parts. Each part is formed by random performances and acts as the testing set once. All the remaining data forms the training set. Certain criteria are selected to assess how well the resulting models predict the testing set. In this chapter, we use the model likelihood to measure the performances of the candidate models. According to the definition of model likelihood in section 2.4.2, a better model for a dataset has a higher likelihood. Here for the convenience of presentation, we show the logarithm scaled likelihood (known as the log-likelihood), unless specified otherwise.

A model selection criterion is a mathematical selector designed for selecting the most appropriate model to fit a set of data. A particular strength of the use of model selection criteria is that all the data can be used for training. However, different model selection criteria have different strengths when used to select models. In this experiment, we use two classical model selection criteria, Akaike's Information Criterion (AIC) and Bayesian Information Criterion (BIC), [Claeskens and Hjort, 2008, ch. 2-3] to test the performance of the resulting models. We provide a more detailed discussion about AIC and BIC in section 2.4.2. The definition of AIC and BIC can be written as:

$$AIC = 2 * o(\theta) - 2 * \text{likelihood} \tag{3.17}$$

$$BIC = log(N) * o(\theta) - 2 * \text{likelihood} \tag{3.18}$$

where $o(\theta)$ represents the number of parameters in the candidate model and the dataset has N samples in total. However, the model selection criteria cannot compare the performances of models with different databases. After the preprocessing, the data of expressive timing are essentially different data as the original performance data are now scaled according to different factors. As a result, the model selection criteria can only be used to compare different settings

of GMMs rather than to compare different standardisation methods.

3.5 Results

3.5.1 Cross-validation tests

In this section, we compare the clustered and non-clustered models in the cross-validation tests. Using the best clustered models, we then discuss which type of covariance matrix and which standardisation methods are the most suitable for clustering expressive timing within a phrase.

First, we compare the clustered and non-clustered models. For clustered methods, we select a GMM with two components to enable a simpler comparison. The results shown in Table 3.1 are negative cross-validation log-likelihood, thus a more negative value means a better model performance.

Neg. Likelihood: Stand. / Model	RR	MR	MVR	LOG
CD models	2.53	-6.44	10.57	-10.02
ND models	1.79	-7.22	9.80	-10.46
$\mathcal{M}^2_{SD}$	1.51	-7.60	9.65	-11.30
$\mathcal{M}^2_{ID}$	-2.69	-8.07	9.40	-11.49
CF models	2.41	-16.24	-0.87	-11.61
NF models	1.57	-16.73	-1.59	-12.14
$\mathcal{M}^2_{SF}$	1.38	-16.85	-1.64	-12.33
$\mathcal{M}^2_{IF}$	-2.81	-17.29	-1.90	-12.97

Table 3.1: Cross-validation tests of non-clustered models, where the statistics are negative log-likelihood per sample and a more negative value means better model performance. RR, MR, MVR and LOG are standardisation methods defined in 3.2, while $\mathcal{M}$ means GMM as defined in section 3.3.1. The definitions of CD, ND, CF, NF are in section 3.3.2.

In Table 3.1, we notice that for the RR, MR and MVR standardisation methods, the constraints for the covariance matrices have major effects on the cross-validation log-likelihood test. In general, F models have better performance than D models. However, under the same type of covariance matrices, the clustered models perform better than the non-clustered models. For LOG standardisation methods, the clustered models outperform non-clustered models regardless of covariance matrix regulation. For comparison between standardisation methods, the MR and LOG methods outperform the other standardisation methods.

Neg. Like. Stand. / Model	RR	MR	MVR	LOG
SD	0.82(32)	-7.90(8)	9.27(16)	-12.36(16)
SF	0.81(16)	-17.03(16)	-1.81(16)	-12.53(16)
ID	-11.20(128)	-8.37(8)	8.94(16)	-12.89 (16)
IF	-6.02(8)	-17.35(2)	-2.05(4)	-13.05 (4)

Table 3.2: The best performance of GMMs under different settings of covariance matrices and standardisation methods. The numbers in brackets besides the negative model log-likelihood are the number of Gaussian components in the best performed models under different settings. A more negative value means a better performance for the resulting model.

In Table 3.2, we show the best performances of different types of GMMs with different standardisation methods engaged. In brackets, we show the number of Gaussian components in the best performed GMM. We confirm the results in Table 3.1 (that $\mathcal{M}_{\mathrm{IF}}$(MR) has the best performance). D models and F models have similar results when the data is LOG standardised. Moreover, in general, with the same conditions for all the other parameters, F models are usually better than D models and I models are usually better than S models. These results suggest that the tempo differences of beats are dependent on each other and the covariance between beats in a phrase changes with the shape of tempo

variations within the phrase.

From Table 3.2, we also see that for different types of GMMs with different standardisation methods engaged, the number of Gaussian components in the best performed models varies within a certain range. In 8 out of 16 cases, the best performing models have 16 Gaussian components, which suggests there are close to 16 clusters for the tempo variations within a phrase in the performance of *Islamey*. We discuss the number of Gaussian components in the best performed models further in section 3.7.

3.5.2 Comparison between cross-validation and the model selection criteria

In this section, we use the model criteria AIC and BIC to evaluate the clustered models. From Table 3.1, the model likelihood in the cross-validation test has clearly shown that the clustered models outperform the non-clustered models, therefore, henceforth we no longer consider non-clustered models. In Table 3.3, we list three parameters for model evaluation: negative cross-validation log-likelihood, AIC and BIC. All the parameters use a more negative number to indicate a better performance of candidate model. The candidate models combine all possible variants of covariance matrices and Gaussian components. The GMMs are denoted as IF, SF, ID, SD in order.

Next, we examine how well the cross-validation test and the model selection criteria are correlated. As we used the negative log-likelihood for measuring the performance of candidate models in the cross-validation test and as the model selection criteria we selected in this chapter is based on a negative model likelihood, the agreement between cross-validation and the model selection criteria should show a strong positive correlation. The measure of correlation we selected is Spearman's rho [Spearman, 1904]. This measurement of correlation is not dependent on the linear relationship between two variables for a strong correlation.

We use Spearman's rho to correlate the model selection criteria (AIC and BIC) and the negative log-likelihood in the cross-validation test which results

	SD			SF			ID			IF		
	X	AIC	BIC	X	AIC	BIC	X	AIC	BIC	X	AIC	BIC
N=2	-7.51	-13.51×10^3	-5.42×10^3	-16.84	-31.96×10^3	-23.74×10^3	-8.14	-14.40×10^3	-6.27×10^3	-17.35	-32.78×10^3	-24.3×10^3
N=4	-7.79	-13.94×10^3	-5.76×10^3	-16.96	-32.12×10^3	-23.81×10^3	-8.31	-14.80×10^3	-6.51×10^3	-17.29	-32.87×10^3	-24.03×10^3
N=8	-7.90	-14.34×10^3	-5.99×10^3	-17.00	-32.29×10^3	-23.80×10^3	-8.37	-15.06×10^3	-6.43×10^3	-17.08	-32.90×10^3	-23.17×10^3
N=16	-7.89	-14.72×10^3	-6.02×10^3	-17.03	-32.44×10^3	-23.60×10^3	-8.31	-15.26×10^3	-5.97×10^3	-16.45	-32.91×10^3	-21.42×10^3
N=32	-7.86	-15.05×10^3	-5.64×10^3	-16.95	-32.52×10^3	-22.97×10^3	-8.02	-15.44×10^3	-4.81×10^3	-14.20	-33.12×10^3	-18.09×10^3
N=64	-7.51	-15.16×10^3	-4.33×10^3	-16.59	-32.52×10^3	-21.55×10^3	-7.46	-15.58×10^3	-2.28×10^3	-8.54	-34.37×10^3	-12.28×10^3
N=128	-6.64	-15.07×10^3	-1.41×10^3	-15.49	-32.14×10^3	-18.35×10^3	-5.78	-15.82×10^3	2.81×10^3	25.85	-42.61×10^3	-6.38×10^3
N=256	-4.19	-14.51×10^3	4.80×10^3	-13.12	-31.15×10^3	-11.70×10^3	-1.28	-17.73×10^3	11.59×10^3	1.06×10^3	-63.85×10^3	0.64×10^3

Table 3.3: The negative cross-validation log-likelihood and the model selection criterion with best standardisation MR applied. The N represents the number of Gaussian components in GMMs. The X represents the negative cross-validation log-likelihood.

	AIC				BIC			
	SD	SF	ID	IF	SD	SF	ID	IF
RR	0.33	0.05	**0.83**	-0.74	**0.95**	**0.98**	**0.83**	**0.81**
MR	-0.26	0.40	-0.83	-1.00	**0.95**	**0.78**	**0.95**	**1.00**
MVR	-0.31	0.12	-0.60	-0.74	**0.90**	**0.83**	**0.83**	**0.76**
LOG	0.40	0.19	-0.57	-0.98	**0.95**	**0.81**	**0.98**	**0.98**

Table 3.4: The correlation between the model selection criteria and the negative cross-validation likelihood. Positive correlations are expected. The bold numbers mean the correlation between the model selection criteria and the negative cross-validation log-likelihood is strong enough to pass a significance test.

in varying the number of Gaussian components in candidate GMMs under the same standardisation methods and the same type of model. For example, if we correlate the first column (X under SD) and the third column (BIC under SD), the resulting correlation shows how well BIC and the cross-validation agree (the value is shown as the sixth column in the fourth row, namely MR-SD under BIC, in Table 3.4). The correlations between the model selection criteria and cross-validation under all circumstances are illustrated in Table 3.4.

From Table 3.4, we notice that the BIC has a strong positive correlation with the negative cross-validation log-likelihood. The numbers shown in bold indicate that the correlation between two variables are strong enough to pass the significance test in statistics [Spiegel and Stephens, 2011, p. 246]. In other words, if a number is shown in bold, the negative cross-validation log-likelihood test has a significant positive correlation with the BIC.

In Table 3.3 and Table 3.4, we can see that the best model in the cross-validation test $\mathcal{M}_{\mathrm{IF}}(\mathrm{MR})$ has the best performance. Moreover, BIC can best predict the model performance when a different number of Gaussian components is employed. The results suggest that the model $\mathcal{M}_{\mathrm{IF}}(\mathrm{MR})$ is the best model for clustering expressive timing in a phrase among the candidate models.

3.6 Application to Chopin Mazurkas

As *Islamey* is a private database and the periodicity of melody in *Islamey* may influence the clustering process, we also apply the proposed experiment to Chopin Mazurkas to investigate whether the conclusion with *Islamey* is still valid. The Mazurka database has been used before and has already been annotated by other researchers, so we can confirm the annotation process does not limit the generality of the proposed experiment. Moreover, as the Chopin Mazurkas have less repetitions than *Islamey*, we can further demonstrate that the tempo variations can be clustered regardless of the repetitions in the candidate piece.

In [Sapp, 2007], Sapp annotated five pieces of Chopin Mazurkas with various numbers of performances. However, the proposed experiment requires that the lengths of phrases in a candidate piece be identical throughout the piece. Amongst the Mazurkas annotated by Sapp, two pieces of Mazurkas — Op.24/2 and Op.30/2 — have identical lengths of phrases throughout the piece. Consequently, we choose these two Mazurkas as the new candidate pieces for analysis. In Mazurka Op.24/2, there are 30 phrases that are 12-beats long and there are 64 pieces of performances in the database. As a result, we have $30 \times 64 = 1920$ samples in this model analysis. Similarly for Mazurka Op.30/2, there are 8 phrases and each phrase is 24-beats long in all 34 performances, so we have $8 \times 34 = 272$ samples in this experiment.

3.6.1 Cross-validation tests

First, we compare the clustered models and non-clustered models in Table 3.5, where we present the performance of the candidate models. Similar to the case of *Islamey* in Table 3.1, we notice that with the same standardisation method engaged and the same restrictions applied to the covariance matrix, the clustered models outperform the non-clustered models for both pieces of Mazurkas.

Next, we compare the best performing model under the proposed clustered models and the proposed standardisation methods. We list the best perfor-

Neg. Likelihood / Stand. / Model	RR	MR	MVR	LOG
CD models	4.03	-6.51	16.76	-10.87
ND models	1.02	-8.75	13.54	-11.59
$\mathcal{M}^2_{SD}$	-0.31	-10.77	11.96	-14.31
$\mathcal{M}^2_{ID}$	-1.04	-10.95	11.42	-15.11
CF models	-1.70	-24.80	-0.12	-22.96
NF models	-4.61	-26.38	-2.82	-24.34
$\mathcal{M}^2_{SF}$	-4.85	-26.55	-2.98	-24.66
$\mathcal{M}^2_{IF}$	-6.94	-27.41	-3.94	-25.42

(a) Op.24/2

Neg. Likelihood / Stand. / Model	RR	MR	MVR	LOG
CD models	3.15	-3.80	15.48	-9.09
ND models	2.50	-4.48	14.67	-9.55
$\mathcal{M}^2_{SD}$	-1.44	-8.06	10.51	-11.60
$\mathcal{M}^2_{ID}$	-2.77	-8.31	10.31	-11.68
CF models	-6.56	-24.29	-5.02	-21.55
NF models	-8.04	-25.08	-5.93	-22.70
$\mathcal{M}^2_{SF}$	-8.57	-25.20	-6.33	-22.94
$\mathcal{M}^2_{IF}$	-12.77	-25.82	-7.04	-23.82

(b) Op.30/2

Table 3.5: The comparison between non-clustered models and clustered models in cross-validation tests. Data are shown in negative log-likelihood and a more negative number indicates a better performing model. The representation of the candidate models are defined in section 3.4.

.

Op.24/2	RR	MR	MVR	LOG
SD	-4.33(128)	-15.34(64)	8.09(16)	-20.32(64)
SF	-6.02(32)	-27.52(32)	-3.86(64)	-25.68(64)
ID	-16.16(128)	-16.04(32)	7.17(64)	-20.98(64)
IF	-14.44(8)	-28.85(8)	-5.29(4)	-26.83(4)

(a) Op.24 No.2

Op.30/2	RR	MR	MVR	LOG
SD	-6.16(64)	-12.29(32)	5.51(64)	-17.43(32)
SF	-9.60(8)	-25.71(16)	-7.50(16)	-23.78(8)
ID	-14.22(16)	-13.21(16)	4.60(16)	-17.50(32)
IF	-16.47(4)	-25.99(4)	-8.09(4)	-24.69(4)

(b) Op.30 No.2

Table 3.6: The best performance of different types of GMMs with different standardisation methods engaged. The numbers in brackets are the number of Gaussian components. A more negative value means a better model performance.

mance of models under different settings of covariance matrices and standardisation methods in Table 3.6a and Table 3.6b. From the results, we can see that the best performance of the proposed models are $\mathcal{M}_{\text{IF}}(\text{MR})$ for both Mazurkas. In general, F models outperform D models and I models outperform S models. Both conclusions agreed with the conclusions we drew with the *Islamey* database.

On the other hand, we noticed that the number of Gaussian components differs from piece to piece in the best performed models when the type of covariance matrix and the standardisation methods are the same. For Mazurka Op.24/2 (Table 3.6a), 6 out of 16 best performing models have 64 components. However, for Mazurka Op.30/2 (Table 3.6b), only 2 out of 16 best performed models have 64 Gaussian components. We show the comparison of the number of Gaussian components in the best performed models for each candidate piece

in section 3.7 in order to investigate the number of Gaussian components in the best performing models.

3.6.2 Comparison between the model selection criteria and cross-validation

	AIC				BIC			
	SD	SF	ID	IF	SD	SF	ID	IF
RR	**0.83**	0.19	**0.86**	-0.69	**0.76**	**0.78**	**0.90**	**0.83**
MR	**0.93**	0.10	0.57	-0.74	**0.67**	**0.71**	**0.67**	**0.90**
MVR	**0.83**	0.19	0.57	-0.67	0.62	0.55	**0.67**	0.21
LOG	0.86	0.33	0.57	-0.83	**0.86**	**0.62**	**0.76**	**0.97**

(a) Op.24 No.2

	AIC				BIC			
	SD	SF	ID	IF	SD	SF	ID	IF
RR	-0.32	-0.96	-0.50	-0.61	**0.96**	**0.89**	0.11	0.36
MR	-0.50	-0.96	-0.68	-0.21	**0.96**	**0.96**	0.17	0.36
MVR	-0.32	-0.71	-0.54	-0.75	**0.89**	**0.68**	0.11	0.28
LOG	-0.32	-1.00	-0.50	-0.21	**0.92**	**0.96**	0.21	0.36

(b) Op.30 No.2

Table 3.7: The correlation between the model selection criteria and the negative cross-validation likelihood. Positive correlations are expected.

Next, we investigate if the model selection criteria can predict the results of the cross-validation tests. In Table 3.7, we show the correlation between the model selection criteria and the negative cross-validation likelihood for both Mazurkas. We find that, in some cases, BIC fails to show a significant correlation with the cross-validation likelihood. However, the model we suggested in the *Islamey* $\mathcal{M}_{\mathrm{IF}}(\mathrm{MR})$ dataset shows the significance of the correlation in both Mazurkas. Thus, we conclude that $\mathcal{M}_{\mathrm{IF}}(\mathrm{MR})$ (Gaussian Mixture Model with

Independent Full Matrix and with Mean Regulation Standardisation method applied) is the best model among the candidate models.

3.7 Discussion

In this chapter, we investigate how mathematical models predict the distributions of expressive timing within a phrase. The results support the following statistical conclusions:

1. Clustered models outperform non-clustered models for predicting tempo variations distribution on the data we tested.

2. The best model in the cross-validation tests is $\mathcal{M}_{\mathrm{IF}}(MR)$ on the data we tested. More generally, the model with full covariance matrices is better than the model with diagonal covariance matrices. The model with independent covariance matrices for each Gaussian component is better than the model that has a shared covariance matrix for each Gaussian components.

3. The number of Gaussian components in the best performing models varies according to the different pieces.

4. Compared with AIC, BIC has a similar result for model selection.

In Table 3.1 and Table 3.5, we can see that if the standardisation method and the covariance matrix are engaged, the clustered models outperform the non-clustered models. From Table 3.1, Table 3.2, Table 3.5 and Table 3.6, we can find a general conclusion that for the data we tested, F models outperform D models. Moreover, on average, the order of standardisation methods is MR, LOG, RR and MVR for the performance of the best performing models. For clustered models, I models outperform S models. For non-clustered models, N models outperform C models. Summarising the above conclusions, according to the data we tested, the model we suggest for modelling expressive timing within a phrase is the Gaussian Mixture Model with Independent Full covariance matrices and the engaged standardisation method is Mean Regulation ($\mathcal{M}_{\mathrm{IF}}(\mathrm{MR})$).

	Islamey	Op.24/2	Op.30/2
N=2	1	0	6
N=4	2	2	2
N=8	3	2	3
N=16	8	1	5
N=32	1	3	0
N=64	0	6	0
N=128	1	2	0
N=256	0	0	0

Table 3.8: The count of the number of times that each number of Gaussian components appeared in the best performing GMMs in the cross-validation likelihood test with each proposed model and standardisation method engaged.

Next we discuss how many Gaussian components are contained in the best performing models. In fact, if we compare Table 3.2 and Table 3.6, the number of Gaussian components in the best performing models differ from piece to piece. In Table 3.8, we count the number of times that each number of Gaussian components appeared in the best performing GMMs in the cross-validation likelihood tests with each proposed models and standardisation method engaged. From the table we can see that the number of Gaussian components in the best performing models differ from piece to piece. The reason for such difference needs further investigation.

To compare the model selection criteria and the negative cross-validation log-likelihood, we use Spearman's rho [Spearman, 1904] to measure the correlation between model selection criteria and the negative cross-validation log-likelihood. Spearman's rho does not demand a linear relationship to have a higher correlation. From the correlation coefficient and the significance tests, the BIC and negative log-likelihood in cross-validation test are more correlated according to Spearmean's rho. By this result, we can assert that the BIC can better predict the model performance in terms of negative cross-validation log-likelihood test.

3.8 Conclusions

In this chapter, we used a model selection test to show that the tempo variations within a phrase can be clustered. We first introduced the pre-processing of the performance data. The smoothing was introduced for approximating human perception and the standardisation was used for removing the speed differences between phrases.

We proposed a few different mathematical models including clustered and non-clustered models. The frameworks of all the models were based on the Gaussian model, which is a widely used model for multivariate distribution. We regulated the covariance matrix and the mean of the non-clustered candidate models. For the clustered candidate models, we proposed a mixture of non-clustered models, GMM, and constricted the covariance matrices in GMM by two ways. We use the Expectation Maximum (EM) algorithm to train the proposed models with the candidate pieces.

To compare the performances of the candidate models, we used cross-validation tests to compare the performances of the proposed models. The database was divided into two datasets: the training and the testing dataset. The proposed models were trained by the training dataset with EM. Then the candidate models were evaluated by testing how likely the testing dataset was observed by the resulting models. This procedure was defined as the cross-validation test. We then evaluated the candidate models by showing how well the model selection criteria predicts the performance in cross-validation tests of the candidate models.

Next, we repeated all the experiments proposed in this chapter for the exemplar piece *Islamey* to two Chopin Mazurkas. The Chopin Mazurkas have a more complicated music structure and possibly more varieties in expressive timing. The validation of the proposed algorithm with the Chopin Mazurkas could be possibly considered as evidence of potential generalisation of the proposed algorithm.

From the results of the cross-validation likelihood tests, the model suggested for clustering expressive timing is the GMM with independent full covariance

matrices and mean regulation standardisation ($\mathcal{M}_{\mathrm{IF}}(\mathrm{MR})$). This result was confirmed by two pieces of Chopin Mazurkas and our private *Islamey* database. It would be interesting to test if this conclusion can be generalised to other databases.

Chapter 4

Model Analysis for Expressive Timing across Phrases

In Chapter 3, we have demonstrated that it is useful to cluster the expressive timing. As we discussed in chapter 2, it would be interesting to investigate how the clusters of tempo variation are chosen by different performers to form expressive performances. In this chapter, we investigate what factors impact the decisions about clusters of expressive timing by performers. In other words, we examine what factors decide the expressive timing of a phrase.

There have been a few attempts at determining suitable expressive timing for a segment of music, especially for music expression synthesis. The rule-based system KTH [Friberg et al., 2006] has a set of rules for synthesis performances, including the consideration of the musical score. However, these rules, according to [Friberg et al., 2006], do not consider the potential impact on tempo variations by the tempo variations in previous phrases. Widmer et al. ([Widmer et al., 2010] and [Tobudic and Widmer, 2003b]) discuss how expression in performed music is formed from musical score. Widmer et al. [Widmer et al., 2010] proposed making use of empirical data to generate per-

formance expression by finding parts in the given musical score that are similar to parts of the training pieces. Moreover, Widmer et al. ([Widmer et al., 2010]) also suggested that a dynamic Bayesian network [Murphy, 2012, p. 631] could enhance the resulting model, in which case, the sequential of tempo variations could modelled besides the impact of the score for expressiveness. Moreover, Todd [Todd, 1992] pointed out that parabolic curves can be used for fitting tempo variations across different levels of a music structure. As a result, the tempo variation throughout a phrase is likely to have a slow-fast-slow gesture. When part of the tempo variation within a phrase is known, we can predict how to use the slow-fast-slow gesture to predict the tempo variation of the remaining parts of the phrase.

From previous works, we conclude that there are two major factors to be considered: the expressive timing in the neighbouring phrase and the position of phrase in score. To investigate the reasoning of using certain clusters of expressive timing, we built up mathematical models that describe three variants for a particular phrase: the cluster used for the previous phrase, the cluster used for the current phrase and the position of the current phrase. We propose three models according to: 1) Todd's model [Todd, 1992], which concerns the expressive timing in the neighbouring phrase only, 2) Widmer's work [Widmer et al., 2010], which concerns the position of the phrase only and 3) Widmer's suggestion [Widmer et al., 2010], which concerns both expressive timing in the neighbouring phrase and the position of the phrase. Moreover, we propose a reference model that considers the cluster used for the current phrase, the cluster used for the previous phrase and the position of the phrase as independent variants. We examine how much better the proposed models are than the reference model for comparison purposes.

Before the discussion about how performers choose clusters of tempo variations for phrases, we need to know how the clusters are defined. Thus we first introduce how the intra-phrase tempo variations are clustered according to the conclusion in Chapter 3. Then we introduce the way to visualise the use of clusters throughout a piece of performance. This visualisation method also

motivates the idea of a hierarchical structure in Chapter 5.

From the visualisation of the use of clusters in expressive performances, we notice that different performers use certain clusters in different ways. To compare the performance of all the models in the experiment, we use cross validation methods again. We use observed data to train the candidate models and then measure how well the resulting models are able to predict unobserved data. Evaluating the performance of the candidate models is a critical problem in this experiment hence we need a proper measurement of performance. The comparison between the candidate models differs from Chapter 3 because the complexity of the candidate models differs from each other. As a result, we need a new measurement for model selection. By demonstrating the equivalence between cross entropy and cross-validation likelihood ([Murphy, 2012, p. 274]) first, we propose a measurement of model selection based on information theory. Such a measurement can give us extra flexibility to measure the complexity of different models. The evaluation we propose in this chapter is called cross entropy ratio, which estimates the efficiency of using observed data to predict unobserved data according the candidate models.

For selecting a proper model, we also need to test the data-size candidate of the propose models. The data-size robustness of a model is defined as how well a model can predict unobserved samples when only very limited samples are used for training [Xu and Mannor, 2010]. A data-size robust model can be trained effectively as it needs only a fairly small amount data for training. To test the data-size robustness of the proposed models, we reduce the availability of the samples for training and used a cross-validation test to examine the performances of the resulting models.

Moreover, as the cluster of expressive timing is adapted from the previous chapter, the database we use in this chapter will be the same as that used in the previous chapter. We also used our self-annotated *Islamey* database and the two public databases of Chopin Mazurkas (Op.24/2 and Op.30/2).

This chapter is organised in the following way. We first introduce the visualisation of the resulting clusters of expressive timing. Then we introduce our four

proposed models in detail and the query likelihood test procedures. We then propose three model selection criteria to test the performance and robustness of the proposed models. Finally, we give possible conclusions for choosing clusters of expressive timing in a piece of performance.

4.1 Tempo Variegation Map

4.1.1 Clustering of expressive timing

To investigate what factors impact the choice of clusters of expressive timing, we first need to make a hypothesis. In this section, we introduce a visualisation tool for observing the choice of clusters of expressive timing. This visualisation tool will help us to propose four candidate models in this chapter.

Before discussing how to visualise the use of clusters of expressive timing, we use the Gaussian Mixture Model (GMM) proposed in Chapter 3 to cluster the expressive timing within a phrase first. First, let us recall that the model we suggested for clustering expressive timing within a phrase is a GMM with Independent Full covariance matrices and with standardisation of the Mean Regulation engaged ($\mathcal{M}_{\text{IF}}(\text{MR})$). Then recalling the definition of GMM with Independent Full covariance matrices ($\mathcal{M}_{\text{IF}}$) in equation (3.16), the mathematical definition of the proposed model is:

$$p(\tau_i) = \sum_{a=1}^{A} \pi_a \mathcal{N}(\tau_i | \mu_a, \mathbf{\Sigma}_a^{full}). \tag{4.1}$$

If we use τ_i^* to represent the cluster that τ_i belongs to, we have

$$\tau_i^* = arg\max \pi_a \mathcal{N}(\tau_i | \mu_a, \mathbf{\Sigma}_a^{full}). \tag{4.2}$$

In other words, the expressive timing within a phrase is clustered into the cluster that has the highest posterior probability to the respective Gaussian component.

Suppose there are A Gaussian components in the GMMs we proposed, then we can use numbers from 1 to A to represent the clusters that are associated with each Gaussian component. If we use a row vector to represent clusters used

by performers throughout a piece of performance. The vector ($\overrightarrow{C_m}$) represents the clusters of expressive timing used throughout the mth performance in the database. For each phrase in the mth performance, we use C_{mn} to represent the cluster used for the nth phrase. As a result, we can write $\overrightarrow{C_m} = (C_{\mathrm{m1}}, C_{\mathrm{m2}}, \ldots)$. Suppose we have a database that has M performances and there are N phrases in each performance, we can use a matrix $\mathbf{C}$ to represent the clusters used by different performers throughout a piece of performance in a database. Thus we have

$$\mathbf{C} = \begin{bmatrix} \overrightarrow{C_1} \\ \overrightarrow{C_2} \\ \vdots \\ \overrightarrow{C_M} \end{bmatrix} = \begin{bmatrix} C_{11}, C_{12}, \ldots, C_{1\mathrm{N}} \\ C_{21}, C_{22}, \ldots, C_{2\mathrm{N}} \\ \vdots \quad \vdots \quad \ddots \quad \vdots \\ C_{\mathrm{M1}}, C_{\mathrm{M2}}, \ldots, C_{\mathrm{MN}} \end{bmatrix}. \tag{4.3}$$

As we have A clusters, $C_{\mathrm{mn}} \in [1, A]$ and $C_{\mathrm{mn}} \in N^+$. Although $\mathbf{C}$ shows the usage of clusters of expressive timing in different phrases across different performances, a matrix is more difficult for observation compared with a diagram. Next we are going to convert matrix $\mathbf{C}$ to a diagram.

4.1.2 Colour assignment for the clusters of expressive timing

A straightforward method to convert a matrix to a diagram involves using a colour matrix. Each element in the matrix is represented by a small coloured block. The colour of each block is chosen according to the value of the element. In our proposed visualisation method, we use this method to convert matrix $\mathbf{C}$. The resulting diagram is called the Tempo Variegation Map (TVM). We use the Chopin Mazurka Op.24/2 to show how these colour assignments help us observation the use of clusters of expressive timing.

As seen in Table 3.6a, we use $\mathcal{M}^{8}_{\mathrm{IF}}(\mathrm{MR})$ to cluster the expressive timing within a phrase in Chopin Mazurka Op.24/2. The resulting centroids of each cluster are shown in Figure 4.1. In this section, we see two different visualisations with two different colour schemes. Each colour scheme selects the colour

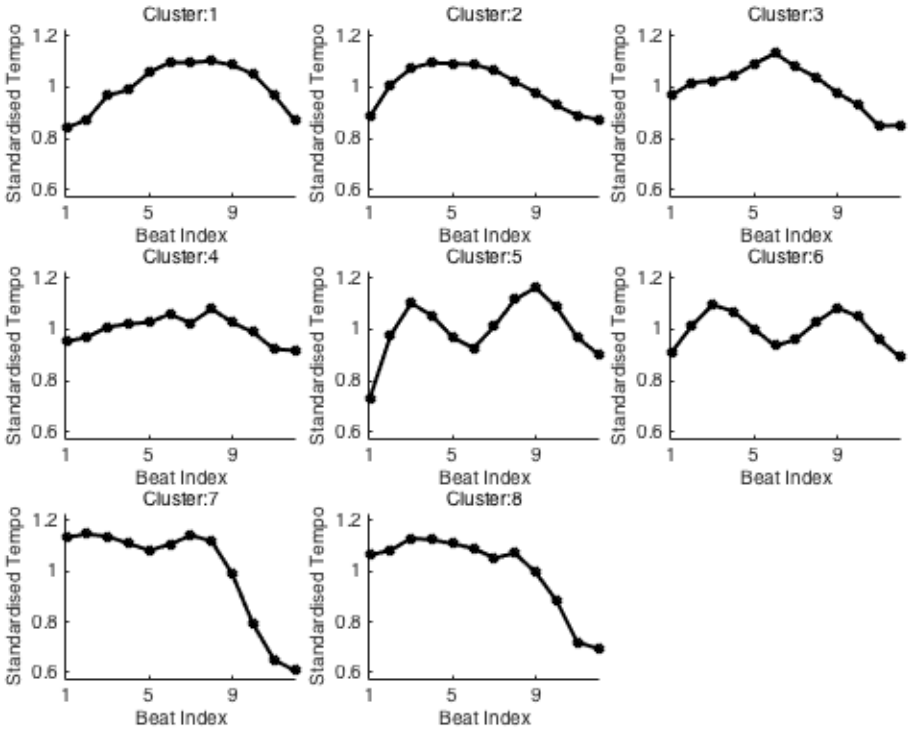


Figure 4.1: Centroids of the clusters of expressive timing in Chopin Mazurka Op.24/2.

according to different criteria. See Appendix B for a detailed description of the colour schemes used.

The criteria we selected for the colour schemes presented are the shape of the centroids and the acceleration rates of the centroids. Figure 4.2 shows the resulting TVM based on the colour scheme concerning the shapes of the resulting centroids. In this figure, centroids with similar shapes are shown with similar colours. Figure 4.3 shows the resulting TVM based on the colour scheme concerning the acceleration rates of the resulting centroids. Green represents centroids slowing down at the end of a phrase, whereas blue represents centroids speeding up at the end of a phrase. There are three parts in Figure 4.2 and Figure 4.3. The top part shows the TVM, the middle part shows the colour of each cluster and the bottom part shows the centroids of clustered expressive timing with the colour assigned. The indexes of clusters in both diagrams

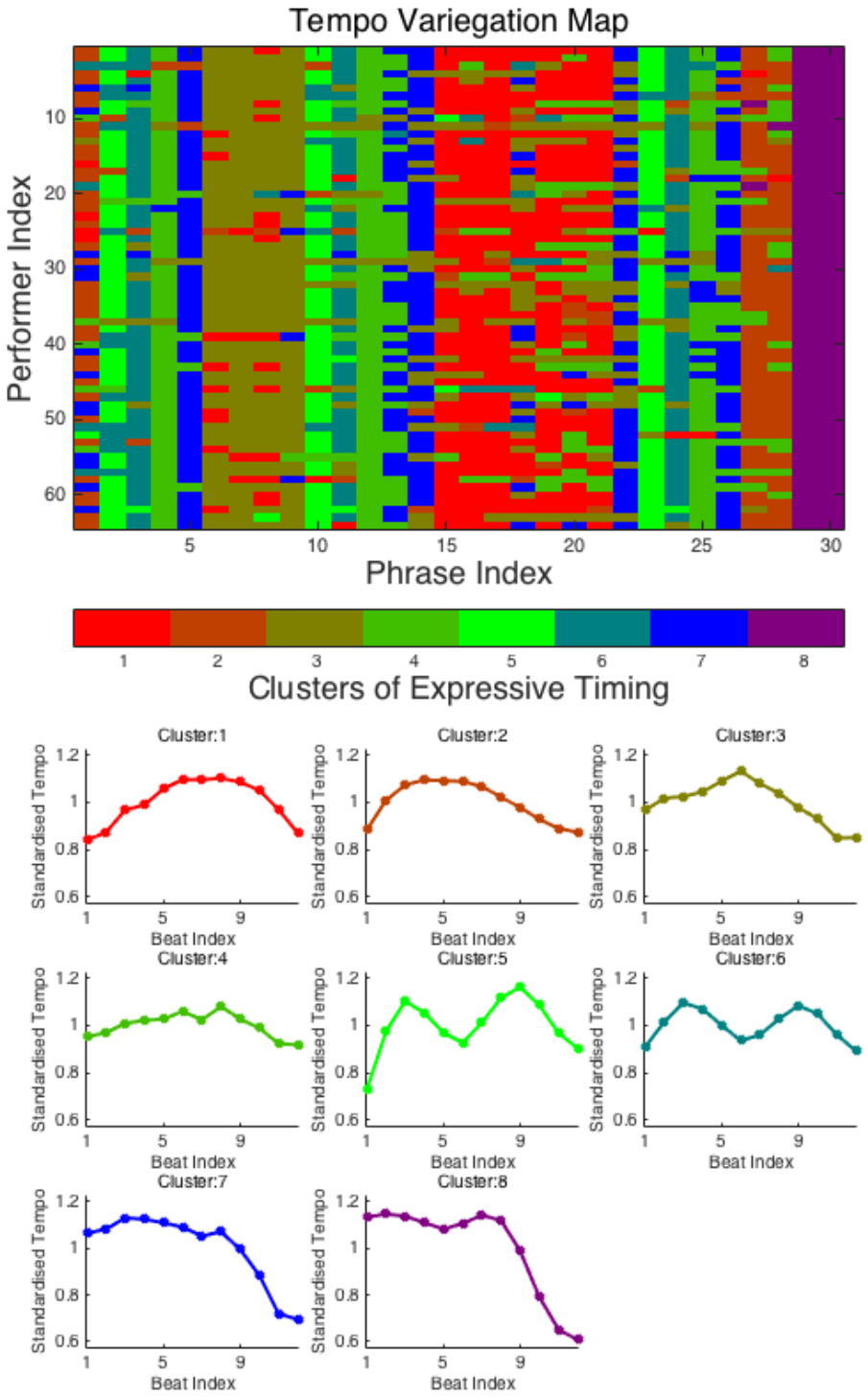


Figure 4.2: Tempo Variegation Map (TVM) and the colours of the clusters of expressive timing, colouring scheme considers the shape of the resulting centroids. The indexes of clusters correspond to cluster index in Figure 4.1.

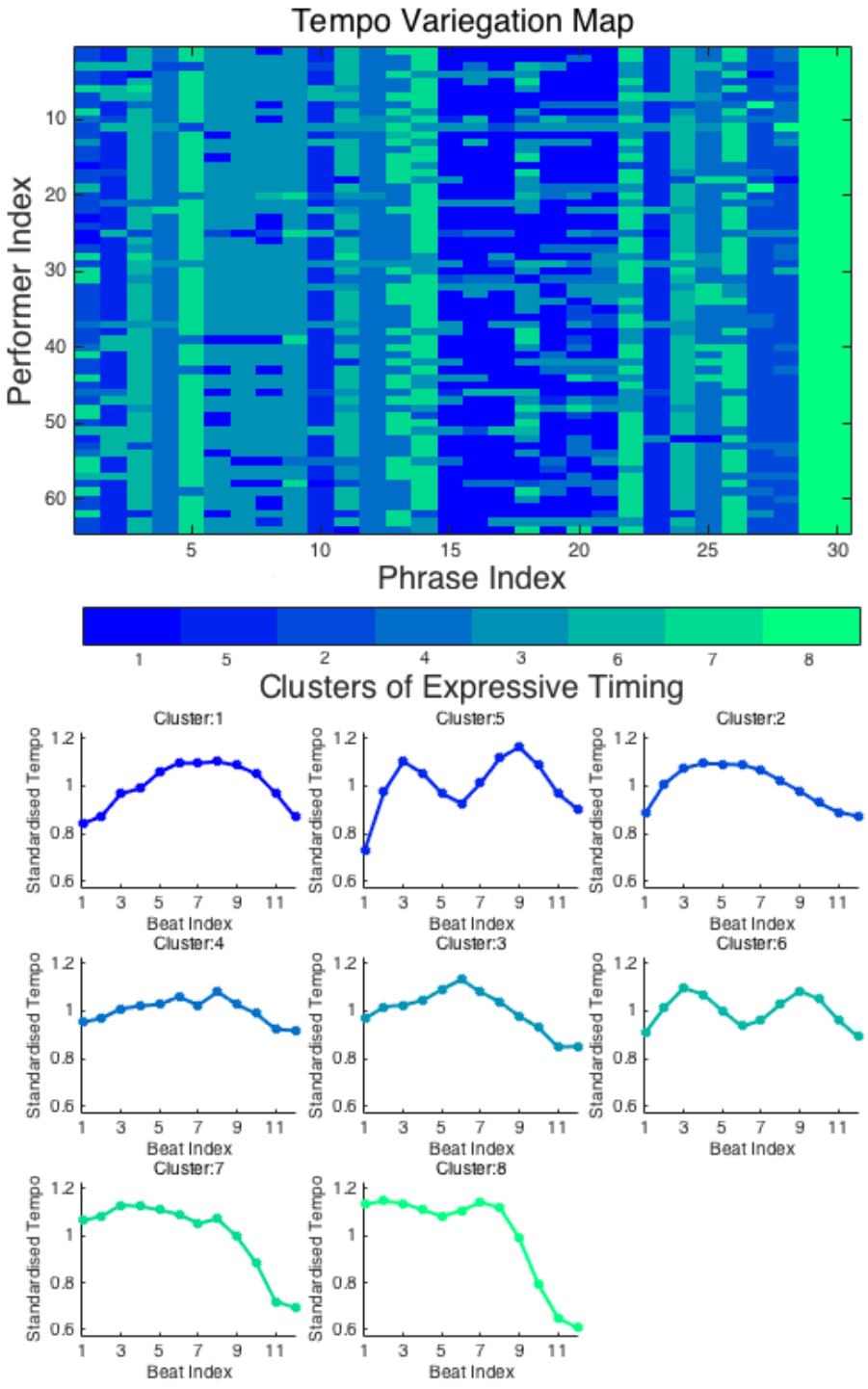


Figure 4.3: Tempo Variegation Map (TVM) and the colours of the clusters of expressive timing, colouring scheme considers the accelerate rate of the resulting centroids. The indexes of clusters correspond to cluster index in Figure 4.1.

correspond to the indexes used in Figure 4.1.

However, because many criteria can be considered to assign different colours to different clusters, it would be interesting to investigate how to assign colours to clusters of expressive timing. With a different colouring scheme, different observations can be made, leading to different musicological hypotheses.

In a TVM, each row represents a piece of a performance, and each block represents a phrase within the performance. By observing TVMs row by row, we can determine how a performer uses different clusters of expressive timing throughout a piece of a performance. Each column in the TVMs represents a specific phrase in a piece of music. If we observe a single column in the TVMs, we can compare how different performers vary their expressive timing for the same phrase. There are several other ways to observe TVMs. In the next section, we will make a few hypotheses based on the observations shown in Figures 4.2 and 4.3.

4.1.3 Observations of TVMs

In this section, a few observations of Figure 4.2 and Figure 4.3 are presented. We first start with Figure 4.2. The colouring scheme in Figure 4.2 represents the shapes of centroids. From the resulting TVMs, we notice that different shapes of centroids are distributed in different positions. For example, between phrase 6 to phrase 9, cluster 3 is commonly used. Cluster 1 is commonly used between phrase 15 to phrase 21. Cluster 2 are the most common cluster in phrase 27 and 28. In phrase 29 and phrase 30, all performers choose cluster 8. With these observations, we can hypothesise the factors that impact the choice of clusters of expressive timing for a phrase.

In Figure 4.4, we show the count of the use of clusters of expressive timing in the Mazurka database for different phrases in Mazurka (Op.24/2). From this diagram, we notice that the use of clusters of expressive timing differs from phrase to phrase. This observation demonstrates a conclusion made by Spiro [Spiro et al., 2010], who showed that the variations of expressive timing differ according to the position in the performance. With this observation in mind,

we propose a model called a Positional Model (PM) that hypothesises that the use of clusters of expressive timing is impacted by the positions of the phrases in a piece of performance.

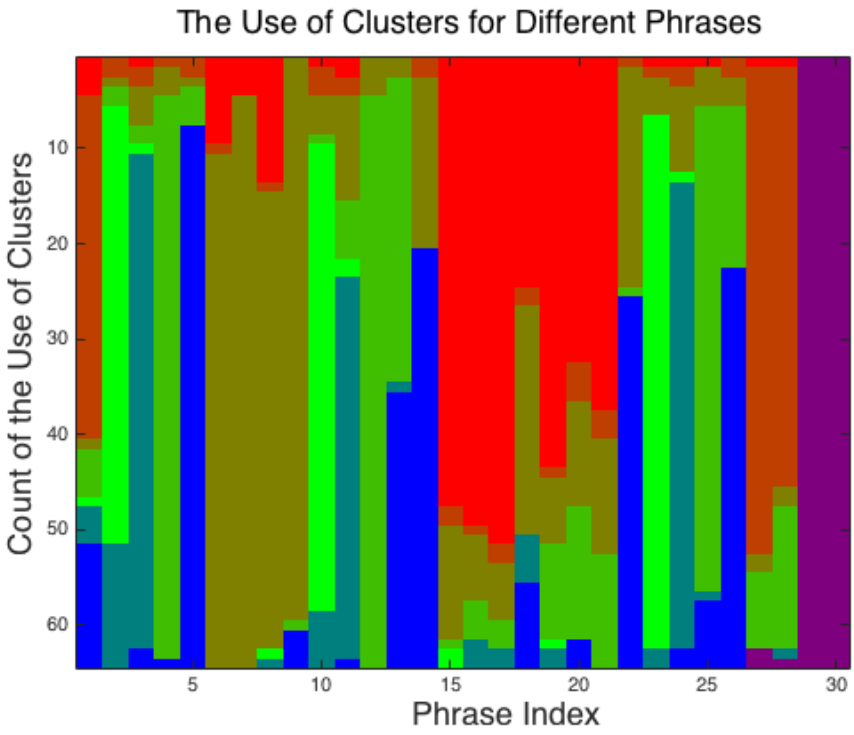


Figure 4.4: The count of the use of clusters of expressive timing in different phrases. The colours of the blocks for counting correspond to the clusters of expressive timing demonstrated in the bottom part of Figure 4.2.

In Figure 4.2, we also find that some clusters are often followed by another. For example, cluster 5 is often followed by cluster 6 throughout the piece. So we want to examine which clusters are used after a particular cluster of expressive timing is used in the previous phrase. In order words, we want to count how many times each colour of blocks appear after a specific colour block. Figure 4.5 shows the distribution of clusters of expressive timing after each cluster of expressive timing. For easier observation, we use the colour of the cluster of expressive timing to represent the count of the use of a respective cluster of expressive timing.

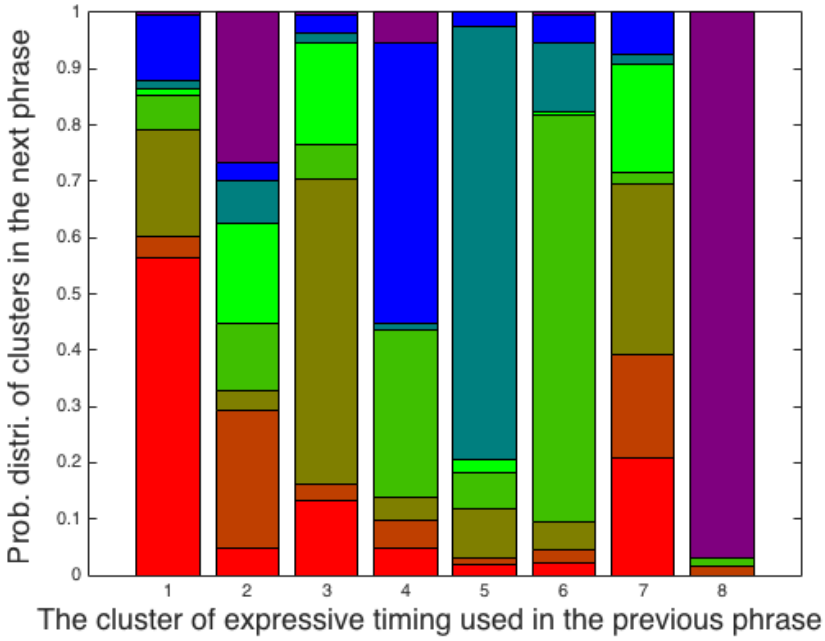


Figure 4.5: The count of the use of clusters of expressive timing after a cluster of expressive timing is used in the previous phrase. The colours of the blocks for counting correspond to the clusters of expressive timing demonstrated in the bottom part of Figure 4.2.

From the digram, we notice that if a cluster is used for a phrase, the distribution of clusters of expressive timing for the next phrase is impacted by the cluster of expressive timing used in the previous phrase. For example, if cluster 8 in Figure 4.5 is used in a phrase, then it is likely cluster 8 will be used again in the next phrase. However, if cluster 7 is used in a phrase, clusters 1, 2, 3, 5 and 7 are likely to be used for the next phrase and cluster 8 is never used. This fact suggests that a cluster of expressive timing used for a phrase impacts on the choice of the cluster of expressive timing in the next phrase. Consequently we propose a candidate model called a Sequential Model (SM) that hypothesises the use of a cluster of expressive timing is impacted by the cluster of expressive timing used in the previous phrase.

Sometimes, a particular cluster (such as cluster 7) can be followed by dif-

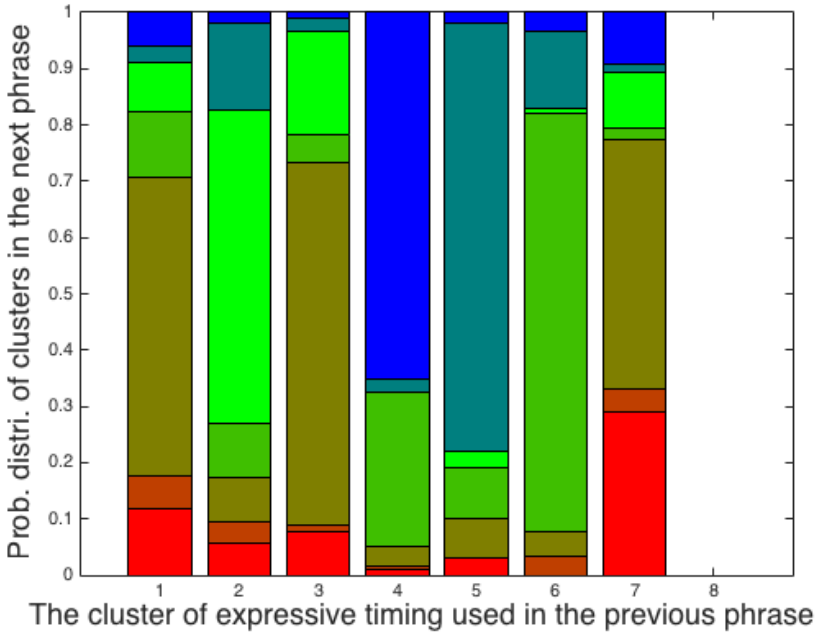


Figure 4.6: The count of the use of clusters of expressive timing in different phrases after a cluster of expressive timing used in the previous phrase in the first half of performance. The colours of blocks for counting correspond to the clusters of expressive timing demonstrated in the bottom part of Figure 4.2.

ferent clusters according to the position of a phrase. For example, cluster 7 is likely to be followed by cluster 3 in phrase 5 whereas in phrase 22, cluster 7 is likely to be followed by cluster 5 instead. In Figure 4.6, if we only analyse the first half part of performance, the distribution of the clusters of expressive timing after each cluster used in the previous phrase is different than the distribution of clusters in the whole performance. For example, if cluster 1 is used in a phrase, another cluster 1 is likely to be followed but cluster 5 is more likely to be used in the next phrase if only the first half of performance is considered. Moreover, cluster 8 does not even appear in the first half of performance. Based on such observations, we build up a Joint Model (JM) that hypotheses in a phrase, the use of cluster of expressive timing is affected by both the cluster of expressive timing used in the previous phrase and the position of the

phrase in the performance. This candidate model was also proposed by Widmer [Widmer et al., 2010].

Based on the observations in Figure 4.2, we hypothesise that the choice of clusters of expressive timing is affected by different factors. Based on the observations in Figure 4.3, which presents the acceleration rates of centroids, we can form a different hypothesis. The results observed in Figure 4.3 may be related to the changes in expressive timing within a phrase. Similar to the results in Figure 4.2, Figure 4.3 shows that different performers use common sets of clusters. We count the number of times each cluster is used in each phrase (Figure 4.7). Unlike in Figure 4.4, Figure 4.7 shows how the expressive timing commonly changes within each phrase.

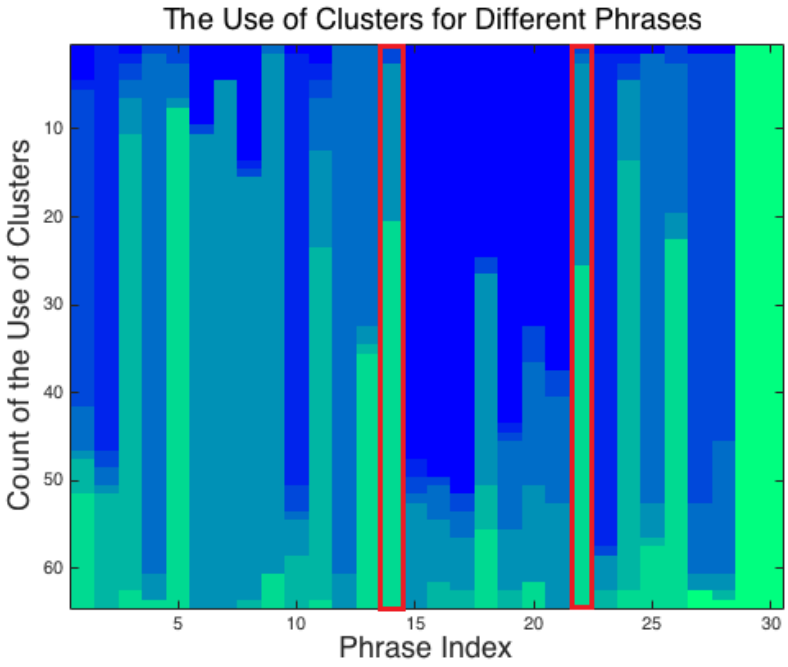


Figure 4.7: The count of the use of clusters of expressive timing at different phrases. The colours of blocks for counting is corresponding to the clusters of expressive timing demonstrated in the bottom part of Figure 4.3.

If we examine Figure 4.7 and the information of music structure provided in the Mazurka database, we find an interesting phenomenon. Except at the be-

ginning and the ending of the performance, there are two important boundaries of music structure marked by the database. The first one is marked at the end of phrase 14, the second one is marked at the end of phrase 22. We mark both phrases in a red frame in Figure 4.7. We noticed that the clusters used by the performers in phrase 14 and phrase 22 are cluster 3, 6 and 7. All used clusters show a deceleration of tempo values as shown in Figure 4.2. Moreover, at the end of the performance, all performers use cluster 8, which has the strongest deceleration at the end of phrase among all clusters. This fact suggests that as they approach a music structure boundary, the performer tends to reduce their tempo. Similarly, the parabolic regression of tempo variations in [Todd, 1992] also shows the deceleration of tempo at the end of a music structure.

In the present chapter, we will examine how the choice of clusters of expressive timing is impacted by different factors. The candidate models are based on the observations of Figure 4.2. We shall make use of the observations of Figure 4.3 in Chapter 5.

4.2 Inter-phrase Expressiveness Models

Before evaluating the proposed models, we give details about the candidate models. The data provided for training the proposed models are the elements in matrix $\mathbf{C}$ (defined in section 4.1.1, namely each row in matrix $\mathbf{C}$ represents the clusters of expressive timing used for a particular position in the performance), which is used to represent the use of clusters of expressive timing in the whole database. There are four candidate models for our analysis.

As in the candidate models of SM and JM, there are two clusters of expressive timing involved: one cluster of expressive timing for the current phrase analysed and another cluster of expressive timing for the previous phrase of the phrase analysed. For simplicity, we use TP2 to represent the cluster of expressive timing used in the phrase analysed and we use TP1 to represent the cluster of expressive timing used in the previous phrase of the phrase we analysed. We use the form $(\mathrm{TP1}, \mathrm{TP2})$ to represent the clusters of expressive timing in the

two successive phrases. In the candidate models of PM and JM, the position feature refers to the position in the music score. We use numbers to index the phrases in a piece of performance according to the order of the phrases. As the candidate models consider different aspects, we investigate the distribution of $p(\text{TP1}, \text{TP2}, \text{position})$ according the candidate models to compare the different candidate models.

Now we introduce the candidate models. Besides the candidate models introduced in section 4.1.3, we also introduce a reference model, namely the Independent Model (IM) that hypothesises both the cluster of expressive timing in the previous phrase of the phrase analysed and the position of the phrase in the performance do not impact on the cluster of expressive timing in the phrases analysed. All candidate models are illustrated in Figure 4.8.

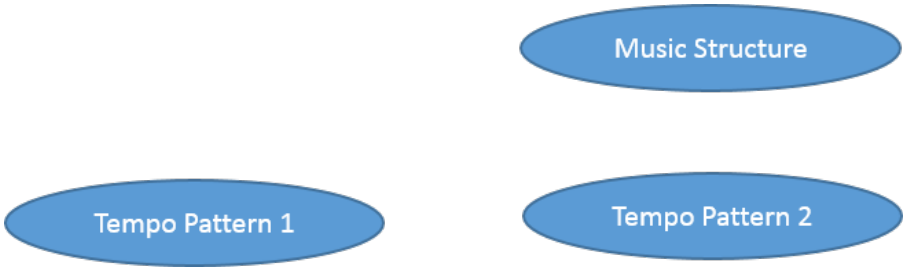
- Independent Model (IM)

  The independent model (as illustrated by Figure 4.8a) assumes that the cluster of expressive timing applied for each phrase is independent from the cluster of expressive timing used in the previous phrase and from the position of the current phrase in a performance. As a result, the distribution of $p(\text{TP}_1, \text{TP}_2, \text{position})$ can be estimated by the multiplication of the distribution of $p(\text{TP}_1)$, $p(\text{TP}_2)$ and $p(\text{position})$. This is because the joint probability distribution of event A and B can be estimated by the multiplication of the probability distributions of event A and event B if events A and B are independent of each other (e.g. $p(A, B) = p(A) \times p(B)$ if $A \perp B$). Namely,

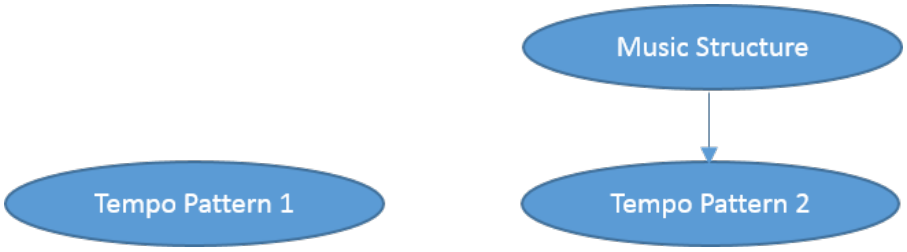
$$p(\text{TP}_1, \text{TP}_2, \text{position}) = p(\text{TP}_1) \times p(\text{TP}_2) \times p(\text{position}). \tag{4.4}$$

- Positional Model (PM)

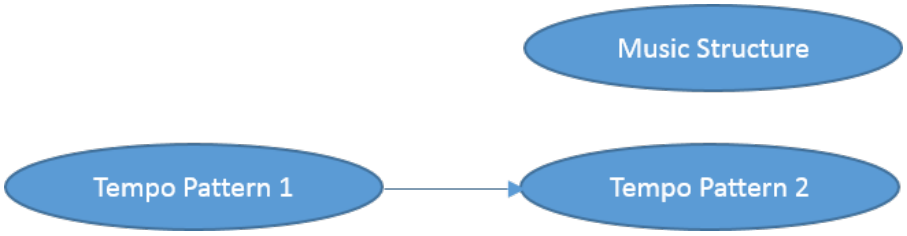
  This model assumes the cluster of expressive timing employed in a particular phrase depends on the music structural information alone. As showed in Figure 4.8b, the cluster of expressive timing used in a phrase is independent of the cluster of expressive timing used in the previous phrase. So the



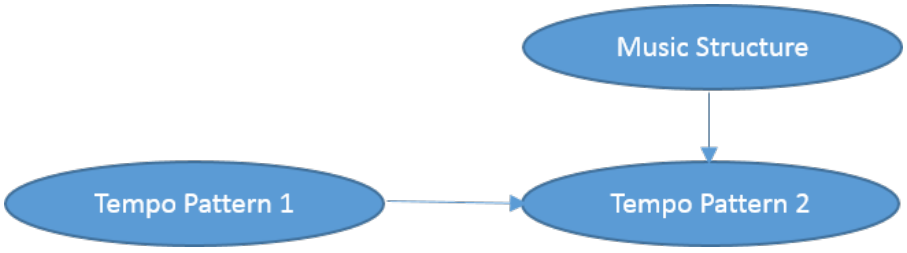
(a) Independent Model (IM)



(b) Positional Model (PM)



(c) Sequential Model (SM)



(d) Joint Model (JM)

Figure 4.8: The candidate models for the usage of clusters of expressive timing

joint distribution of $p(\mathrm{TP}_1, \mathrm{TP}_2, \mathrm{position})$ can be estimated by the multiplication of the probability distribution of $p(\mathrm{TP}_1)$ and the joint probability distribution of $p(\mathrm{TP}_2, \mathrm{position})$, e.g.

$$p(\mathrm{TP}_1, \mathrm{TP}_2, \mathrm{position}) = p(\mathrm{TP}_1) \times p(\mathrm{TP}_2, \mathrm{position}). \tag{4.5}$$

- Sequential Model (SM)

  As can be seen in Figure 4.8c, this model assumes that the cluster of expressive timing used for a particular phrase depends on the previous tempo pattern and thus is independent of the position of the phrase in a performance. The joint distribution of $p(\mathrm{TP}_1, \mathrm{TP}_2, \mathrm{position})$ can be estimated as the multiplication of the joint probability distribution of the clusters of expressive timing in two successive phrases $p(\mathrm{TP}_1, \mathrm{TP}_2)$ and the probability distribution of the position in a piece of score $p(\mathrm{position})$. In other words,

$$p(\mathrm{TP}_1, \mathrm{TP}_2, \mathrm{position}) = p(\mathrm{TP}_1, \mathrm{TP}_2) \times p(\mathrm{position}). \tag{4.6}$$

- Joint Model (JM)

  The joint model assumes that the choice of the cluster of expressive timing for a particular phrase depends on both the cluster of expressive timing used in the previous phrase and the position of the current phrase in the performance. The structure of this model is shown in Figure 4.8d. The probability mass distribution can be estimated by the frequency count of the clusters of expressive timing used in the two successive phrases and the position of the phrase jointly:

$$p(\mathrm{TP}_1, \mathrm{TP}_2, \mathrm{position}) = \textit{histogram count}(\mathrm{TP}_1, \mathrm{TP}_2, \mathrm{position}). \tag{4.7}$$

4.3 Model Evaluation

In this section, we introduce the experiment we proposed for evaluating the proposed models and the measurement we proposed. In the past research [Madsen and Widmer, 2006], Madsen and Widmer converted the expressive timing patterns to a string. This fact suggests that the query likelihood for a language model ([Manning et al., 2009, ch. 12]) can be used for analysing expressive timing. In the query likelihood test, we first train the candidate models with the elements in matrix $\mathbf{C}$. Then we use the elements in matrix $\mathbf{C}$ to query the probability of observing matrix $\mathbf{C}$, which is called a query likelihood [Manning et al., 2009, ch. 12]. To avoid the overfitting problem, similar to Chapter 3, we apply cross-validation for testing the performance of the candidate models. The training dataset in the cross-validation test is formed by selecting different columns in matrix $\mathbf{C}$ (defined in section 4.1.1). However, as the complexity of the candidate models are different to each other, but this is not considered by the query likelihood, we need to find a measurement of the model performance that considers the complexity of the candidate models. We will introduce the criteria for model selection in the second part of this section.

4.3.1 Query likelihood test

The general idea of a query likelihood test [Manning et al., 2009, ch. 12] is to use a certain number of samples to train a candidate model and then to query the probability of observing either the same or different datasets according to the resulting model. Particularly in this chapter, we also use five-fold cross-correlation. We firstly use four-fifths of the data to train the candidate models and then the remaining one-fifth of the data is used to query the probability to be observed (called query likelihood in short).

One of the major problems of the query likelihood test is the zero probability problem, where, if there are no samples for a particular type of sample in the dataset, there is a "0" probability point in the resulting models, regardless of how the candidate models are designed. However, if there is such a sample that

does not exist in the dataset, the probability of observing the testing dataset for all the candidate models become 0, which makes the resulting performance of all the candidate models become equal, thus it is impossible to select the best model amongst the candidate models. This is known as the zero probability problem [Murphy, 2012, p. 79].

To solve the zero probability problem, we use a Bayesian estimation in the training process [Koller and Friedman, 2009, p. 733]. The Bayesian estimation allows the resulting models to rely on both the training dataset and a prior probability distribution. In our experiment, we use the "add-one" smoothing [Murphy, 2012, p. 79], which uses Dirichlet distribution with n samples and an alpha factor $\alpha = 1$, where all points counted for probability mass have a uniform distribution as the prior probability distribution. Unlike the Maximum Likelihood Estimation (MLE) method [Koller and Friedman, 2009, p. 717], which is purely reliant on the training set, Bayesian estimation can avoid the zero probability problem when there are some samples that exist in the testing dataset but not in the training dataset.

4.3.2 Model selection criteria

In this section, we propose some model selection criteria. As we discussed, the model selection criteria in this section should consider the complexity of the candidate models. First, we show the equivalence between some measurements in information theory and query likelihood. Then we propose the model selection criteria used in this experiment.

To start with, we present a process of calculating a query likelihood. We use $\{a_{t-1}, a_t, b_t\}$ to represent a sample t in the dataset of a query likelihood test that uses the cluster a_{t-1} of expressive timing in the previous phrase (TP1) and the cluster a_t of expressive timing in the current phrase (TP2) at position b_t. A dataset P containing n samples can thus be represented as a set $\{\{a_1, a_2, b_2\}, \{a_2, b_3, c_3\}, ...\{a_n, a_{n+1}, b_{n+1}\}\}$. If the samples in this dataset correspond to a piece of performance, the first sample corresponds to the second phrase in the performance, as there must be a previous phrase for a sample. As

a result, there are $n+1$ phrases in the performance. In other words, the first phrase of a performance is not taken into consideration in the training process as there is no "previous phrase". Now suppose there are n samples in the testing dataset. We use q_{ijk} to represent the probability that TP1 uses cluster i of expressive timing, while TP2 uses tempo pattern j and TP2 represents phrase k ($a_{t-1} = i, a_t = j, b_t = k$) in the training dataset; the query likelihood Q_T is then:

$$Q_T = \prod_{t=1}^{n} q_{a_{t-1}a_t b_t}. \tag{4.8}$$

In practice, to ensure the precision of calculation we usually transfer the multiplication of probability to a sum of the log-scale probability in order to calculate the average log-query likelihood per sample $LogQ_T$:

$$LogQ_T = \sum_{t=1}^{n} log_2(q_{a_{t-1}a_t b_t}). \tag{4.9}$$

To make the query likelihood of different sizes of the testing set comparable, we average the query likelihood on a per sample basis. Furthermore, if we use p_{ijk} to represent the probability of $\{a_{t-1} = i, a_t = j, b = k\}$ and there are n samples in the dataset, the number of samples ($\{a_{t-1} = i, a_t = j, b = k\}$) is $n * p_{ijk}$. Thus, if we have A clusters of expressive timing and there are K phrases in a piece of performance, the average log-query probability can be presented as:

$$\overline{LogQ_T} = \frac{1}{n} \sum_{i=1}^{A} \sum_{j=1}^{A} \sum_{k=1}^{K} \log_2(q_{ijk}) * n * p_{ijk} = \sum_{i=1}^{A} \sum_{j=1}^{A} \sum_{k=1}^{K} p_{ijk} \log_2(q_{ijk}). \tag{4.10}$$

Cross-entropy is a concept in information theory [Murphy, 2012, p. 249]. If we are using a model Q, whose probability distribution is $\{q_1, q_2, ...q_n\}$ to encode a series of symbols P, whose probability distribution is $\{p_1, p_2, ..., p_n\}$, the cross-entropy is how many bits on average are required to code a symbol in P. The cross-entropy is defined as:

$$H_{cross} = -\sum_{i=1}^{n} p_i \log_2(q_i). \tag{4.11}$$

Comparing (4.10) and (4.11), if we use a single letter i to replace the subscription ijk, the average query log-likelihood has the same form of cross-entropy with only a sign difference. The cross-entropy shows how many bits on average represent a symbol in a testing dataset P with a coding system from a training dataset Q. However, as the difficulties of encoding the testing dataset P varies with the entropy of the testing set P, it is fair to compare how many extra bits are required for coding the testing dataset P with the training dataset Q compared with the optimised coding scheme designed for P, which is the difference between the cross-entropy and entropy of P,

$$H_{cross} - H_P = \sum_{i=1}^{n} \{p_i log_2(q_i) - p_i log_2(p_i)\} = \sum_{i=1}^{n} p_i log_2(\frac{q_i}{p_i}). \tag{4.12}$$

The difference between the cross-entropy and testing set entropy has the same form as the KL divergence. The KL divergence [Murphy, 2012, p. 58] from a probability distribution Q to a probability distribution P is defined as:

$$KL_{Div}(P, Q) = \sum_{i=1}^{n} p_i log_2 \frac{q_i}{p_i}. \tag{4.13}$$

The KL divergence, or relative entropy, is a measure of how different a probability distribution Q is when compared to a probability distribution P. The KL divergence is non-negative, but it is not a strict distance metric because $KL_{Div}(P, Q) \not\equiv KL_{Div}(Q, P)$.

The KL divergence shows how many extra bits are required when using a training dataset Q to encode a testing dataset P. This considers the complexity of the testing set.

The KL divergence considers the complexity of the training dataset and assumes the complexity of the tasks for different models are similar. However, in this chapter, the candidate models have different dependencies and thus the complexity of the training processes differ from each other. We need a model selection criterion that normalises the complexity of the training dataset. We propose evaluating the resulting models by the cross-entropy ratio Δ. The cross-entropy ratio represents what data cost is incurred when we use Q to encode P

compared to using an optimised coding system P, e.g.

$$\Delta = \frac{KL_{Div}}{H_P}. \tag{4.14}$$

4.4 Results

In this section, we show how the candidate models perform in the query likelihood test and we then discuss the robustness of the models. The data we used for training the models is effectively based on the TVM. The GMM used to obtain the TVMs is $\mathcal{M}_{\text{IF}}(\text{MR})$, which is the best model in Chapter 3. The number of Gaussian components varies from piece to piece, and is chosen according to the best results in the cross-validation test in Chapter 3. So the TVMs we used in this chapter are the best results of the GMMs that performed best in the cross-validation tests in Chapter 3 for each testing piece.

The database we use is the *Islamey* database and two Chopin's Mazurkas (Op.24/2 and Op.30/2) in the Chopin Mazurka dataset, which is the same as in Chapter 3. In Table 4.1, we list the performance of the candidate models according to three proposed model selection criteria. The models used for clustering the expressive timing within a phrase are: $\mathcal{M}^2_{\text{IF}}(\text{MR})$ (for *Islamey*), $\mathcal{M}^8_{\text{IF}}(\text{MR})$ (for Chopin Op.24/2) and $\mathcal{M}^4_{\text{IF}}(\text{MR})$ (for Chopin Op.30/2).

For all three of the model selection criteria in Table 4.1, a smaller value indicates a better model. According to the data shown in Table 4.1, the best performing model is JM. SM is the second best model, with marginal improvements over IM, while PM is the worst performing model amongst all the candidate models. All three model selection criteria reach an agreement that JM is the best model amongst the candidate models.

Next, we show the data-size robustness of the candidate models. The data-size robustness of our models is defined as the performance reduction of each model in response to the size reduction of the training dataset. To evaluate the impact of the number of samples for training, we appoint different numbers of performances to form the training dataset. For a specific number of train-

Criterion \ Model	IM	PM	SM	JM
Cross-Entropy	7.17	7.64	7.06	**6.80**
KL Divergence	0.97	1.45	0.86	**0.61**
Cross-Entropy Ratio	0.16	0.23	0.14	**0.10**

(a) *Islamey*

Criterion \ Model	IM	PM	SM	JM
Cross-Entropy	10.63	13.31	9.69	**7.74**
KL Divergence	4.22	6.90	3.24	**1.36**
Cross-Entropy Ratio	0.66	1.08	0.51	**0.20**

(b) Chopin, Op.24/2

Criterion \ Model	IM	PM	SM	JM
Cross-Entropy	5.80	6.60	5.60	**4.92**
KL Divergence	1.69	2.49	1.49	**0.81**
Cross-Entropy Ratio	0.41	0.60	0.36	**0.19**

(c) Chopin, Op.30/2

Table 4.1: Comparison between the models that predict the use of clusters. All three model selection criteria use a smaller value to indicate a better model performance. IM, PM, SM and JM are defined in section 4.2. The bold value indicates the best performance of the candidate models.

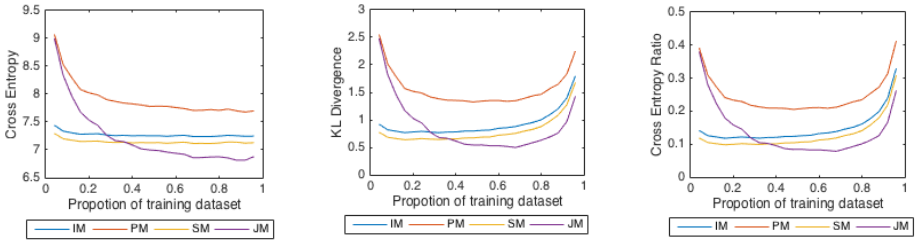
ing performances, we randomly selected the training piece for several times to remove possible bias.

We list the robustness test results of the three model selection criteria for the three test pieces in Figure 4.9, Figure 4.10 and Figure 4.11. In all the figures, the x-axis represent the number of samples used for training. The left end of the axis represents the case where very limited data are used for training the candidate models. The y-axis in this diagram represents the performances of the candidate models. A higher value represents a worse performance of models. So a data-size robust model should be shown as a horizontal line in Figure 4.9, Figure 4.10 and Figure 4.11. On the other hand, a less size-data robust model should be shown as a curve whose left-hand raise rapidly.

According to Table 4.1, JM is the best model. So we firstly discuss the data-size robustness of the JM. We notice that JM is the least data-size robust model amongst the candidate models, regardless of the candidate pieces. For *Islamey*, the JM is beaten by other models when less than about 30% of the performances are used for training. The data-size robustness of JM improves for both Chopin Mazurkas, where JM outperforms the other models when more than 5% of the performances are used for training.

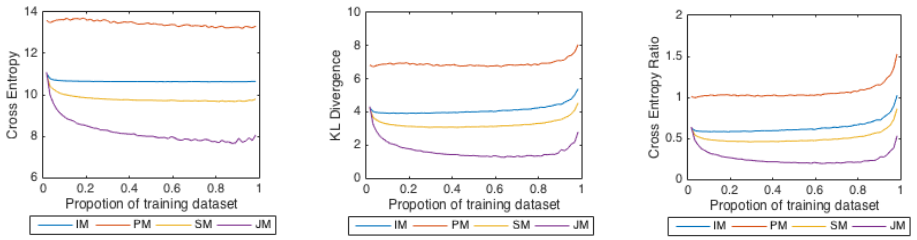
The SM model is more data-size robust than the JM. SM is the best model for *Islamey* when less data ($< 40\%$) is available for training. In the two Chopin Mazurkas, the model selection criteria show that SM is more robust than the JM. The IM is even more data-size robust than the SM. For the two Chopin Mazurkas, the IM outperforms other models when there are severely less data for training (only one piece of performance is used for training).

As a conclusion, we suggest JM to be the model that can best decide the use of clusters of expressive timing. In extreme cases, if there is a very limited amount of data, we suggest SM should be used instead of JM.



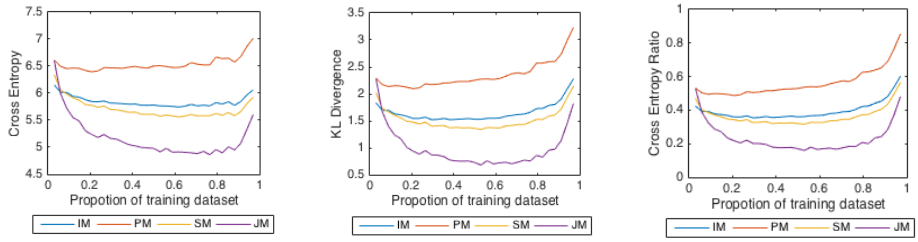
(a) Cross-Entropy (b) KL Divergence (c) Cross-Entropy Ratio

Figure 4.9: Inter-phrase tempo pattern usage modelling comparison for the *Islamey* database. A smaller value indicates a better model.



(a) Cross-Entropy (b) KL Divergence (c) Cross-Entropy Ratio

Figure 4.10: Inter-phrase tempo pattern usage modelling comparison for the Chopin Op.24/2 database. A smaller value indicates a better model.



(a) Cross-Entropy (b) KL Divergence (c) Cross-Entropy Ratio

Figure 4.11: Inter-phrase tempo pattern usage modelling comparison for the Chopin Op.30/2 database. A smaller value indicates a better model.

4.5 Discussion

In this section, we discuss the experiment results from three different aspects. We first compare the three model selection criteria and then discuss the performances of the candidate models in order to make a conclusion about modelling the usage of inter-phrase tempo patterns. Finally we discuss the data-size robustness of the candidate models.

4.5.1 Model criteria comparison

In Figure 4.9, Figure 4.10 and Figure 4.11, the KL divergence and the cross-entropy ratio share a similar change with the change of the training dataset size. The models perform worse at both ends of the curves than in the middle part of the curves when too less or too much data forms the training dataset. The reasons that a very small or a very large training dataset results in bad model performance vary. If there is a very small training dataset, the data we used for training may not have the same distribution of the whole dataset we have, thus the resulting model is not necessarily generalised to the unobserved data or the testing dataset. On the other hand, if we use a very large training dataset, there are less data available to form the testing dataset, but the testing dataset may not have the same distribution as the whole dataset we have. As a result, even if the model we obtained from the training dataset can be generalised, the model selection criteria may indicate a worse model due to the different distribution of the testing dataset.

The cross-entropy hardly shows increase of model selection criterion when a larger training dataset is used, as the cross-entropy does not consider the complexity of the testing dataset. Comparing Figure 4.9a with Figure 4.9b and Figure 4.9c, cross-entropy fails to increase with the drop in model performance for the *Islamey*. Even for the two Chopin Mazurkas, cross-entropy shows a weak response to the model performance when larger training datasets are used.

When we choose the training performances randomly to form the training dataset, there is no way to ensure that all styles of expressive timing are balanced

in the selection. Considering we only repeat the experiment fairly small number of times (at most 100 times, but the possible formation of the training could be larger than 10^{16} according to different databases), it is possible that there is a lack of expressive style balance for the training dataset we randomly formed. As a possible result, the curves (Figure 4.9b, Figure 4.10b and Figure 4.11b) representing KL divergence reflect some small rapid changes. On the other hand, we find that the curves of the cross-entropy ratio are usually more "smooth" than the KL divergence curve. This fact suggests that the cross-entropy ratio can help to reduce the randomness effects as the cross-entropy ratio measures how much percentage of coding length is required rather than how much extra coding length is required (which is measured by the KL divergence).

4.5.2 Model performance

Although the model selection criteria we used evaluate the candidate models from different aspects, an agreement is reached by different model selection criteria. If there are enough training data, the JM outperforms the other models. SM in general is worse than JM but is more data-size robust than JM. IM is the most data-size robust model and if there are severely little data available for training, IM outperforms the other models for the three testing pieces in our experiments. The PM model, regardless of data-size robustness, is the worst model amongst the proposed models.

The rank of candidate models suggests that for inter-phrase expressive timing patterns of a particular phrase, the position in the music score must be considered with the expressive timing in the previous phrase. If we generalise the case a little bit, it is possible to assert that a reasonable model for inter-phrase expressive timing should consider both the cluster used in the previous phrase and the position of the phrase. Particularly, the cluster of expressive timing used in this phrase may be more decisive than the position of the phrase, since the position of phrase may not work well as an independent factor.

4.5.3 Data-size robustness of the models

The data-size robustness of the proposed models varies according to the different pieces selected. From the respect of mathematics, we expect that a model with more parameters would be less data-size robust. However, in this experiment, we find this general mathematical principle does not work as expected. Assume we have n Gaussian components for clustering the intra-phrase expressive timing and there are m phrases, if we use $o(\mathcal{M})$ to present the number of parameters in model $\mathcal{M}$, the number of parameters in the proposed models are:

$$o(\text{IM}) = m; \tag{4.15}$$

$$o(\text{PM}) = mn; \tag{4.16}$$

$$o(\text{SM}) = n^2; \tag{4.17}$$

$$o(\text{JM}) = mn^2. \tag{4.18}$$

Consider the GMM we used for intra-phrase expressive timing clustering, we use Table 4.2 to show the number of parameters in all the candidate models for all the testing pieces we used. According to the way we define data-size robustness in this chapter, we use the ratio between the left end (unless specified) and the minimum point of the cross-entropy ratio to measure the robustness of the candidate models. The left end of the curves represents effectively the case that only 1 piece of performance is used as the training dataset. This measure is shown in Table 4.3. As the Chopin Op.30/2 has fewer phrases than the other piece, there are fewer training samples available. We also show the case that four performances are selected to form the training dataset in Table 4.3 for the purpose of comparison.

In Table 4.3, the data-size robustness of the proposed models differs from piece to piece. Although the smallest training dataset of *Islamey* has the most

Pieces	Model	Number of Phrases	IM	PM	SM	JM
Islamey	$\mathcal{M}_{\text{IF}}^{2}(\text{MR})$	40	2	80	4	160
Chopin Op.24/2	$\mathcal{M}_{\text{IF}}^{8}(\text{MR})$	30	8	240	64	1920
Chopin Op.30/2	$\mathcal{M}_{\text{IF}}^{4}(\text{MR})$	8	4	32	16	128

Table 4.2: The number of parameters in all the candidate models for the different testing pieces.

Pieces	Training Performance/Samples	IM	PM	SM	JM
Islamey	1/40	1.20	1.91	1.21	4.86
Chopin Op.24/2	1/30	1.09	1.02	1.37	3.33
Chopin Op.30/2	1/8	1.20	1.09	1.48	3.31
Chopin Op.30/2	4/32	1.07	1.02	1.17	1.80

Table 4.3: The measures of data-size robustness for all candidate samples. A smaller number indicates a more robust model. The training performance/samples represents the number of data samples in the smallest training dataset.

training samples, the resulting models are less data-size robust. This fact suggests that the some pieces are harder to model than other pieces. In this case, training for *Islamey* is harder than for the two Chopin Mazurkas because the model performs as less data-size robust when the number of training samples is about the same.

If we observe the rank of model complexity in Table 4.2 and the model robustness in Table 4.3, we notice that the rank of model complexity and model robustness are not necessarily the same. For example, the robustness ranks of PM (1) and SM (2) for Chopin Op.24/2 are higher than their complexity ranks (2 and 3 respectively). This fact suggests that the position of the phrase in the music score and the expressive timing in the previous phrase improve the modelling process for certain pieces, although our experiments show that the position of the phrase only decides the cluster of expressive timing for a phrase

joint with the cluster of expressive timing in the previous phrase.

4.6 Conclusions

In Chapter 3, we suggested a model to cluster the expressive timing throughout a phrase. If we use the model we proposed to cluster the expressive timing and we use colour blocks to visualise the distribution of the clusters, we obtain a diagram called a Tempo Variegation Map (TVM). By examining the TVM, we find there are two possible factors affecting the distribution of the cluster of expressive timing: the position of the phrase and the expressive timing in the previous phrase. In this chapter, we investigated how these two factors affect the choice of cluster of expressive timing for a particular phrase in a performance.

We built four Bayesian graphical models to explore the possible dependencies of tempo on music structural information and temporal information: the Independent Model (IM), the Positional Model (PM), the Sequential Model (SM) and the Joint Model (JM). The independent model assumes that the expressive timing in a phrase is independent to the expressive timing in the previous phrase and the position of the phrase. The positional model assumes that it is the position of the phrase that influences the choice of the cluster of expressive timing. The sequential model assumes that the expressive timing in a phrase is effected by the expressive timing in the previous phrase. The joint model assumes the choice of tempo pattern to be employed is affected by both the expressive timing in the previous phrase and the position of the phrase.

To evaluate the performance of our candidate models we used observed data, which contain the joint distribution of sequential tempo pattern pairs and music structure, to query the probability of appearance of the data unobserved by the candidate models. This method is similar to the query model likelihood test in language model evaluation (Chapter 12, [Manning et al., 2009]).

Unlike the model evaluation in Chapter 3, the complexity of the four candidate models differ from each other. In this chapter, we used cross-entropy, Kullback-Leibler divergence and cross-entropy ratio to evaluate the candidate

models. These model selection criteria can be defined from coding length principles in information theory.

We used these model selection criteria to test the data-size robustness of the proposed samples. We varied the availability of training samples to test the data-size robustness of the candidate models. We also vary the availability of the number of tempo patterns to test the capacity of the proposed models. We also overviewed how the intra-phrase tempo variation clustering impacts the inter-phrase expression modelling process.

In this chapter, we used query likelihood test to compare four different Bayesian graphical models for inter-phrase expressive timing. The candidate models considered different dependencies of expressive timing on the position of the phrase and the expressive timing in the previous phrase. We proposed three different model selection criteria then compared the models performances.

The JM model, which considers both the position the phrase and the expressive timing in the previous phrase for deciding the expressive timing in the current phrase, outperformed the other models in terms of model performance. Moreover, the model assuming that the expressive timing of a phrase depends on only the position of the phrase performed dramatically worse than the other models. This fact suggests that the position of the phrase may only affect expressive timing with a consideration of the expressive timing in the previous phrase.

We also tested the data-size robustness of the proposed models in order to consider the case where only very limited data are available for training. The results show that the effects of the position of the phrase and the expressive timing in the previous phrase vary from piece to piece. This fact suggests that the position of the phrase and the expressive timing in the previous phrase also affects different pieces of music differently.

Chapter 5

The Hierarchical Structure of Expressive Timing

In Chapter 3, we demonstrated that expressive timing within a phrase can be clustered. However, as past works have shown, the clustering analysis can be applied to various lengths of music. For example, the unit used by Spiro et al. [Spiro et al., 2010] is a bar; the unit used by Madsen and Widmer [Madsen and Widmer, 2006] is half an bar. According to Todd [Todd, 1992], the expressive timing within different units may have hierarchical relationships. Tobudic and Widmer [Widmer and Tobudic, 2003] proposed a hierarchical system that synthesises expressiveness. In this chapter, we propose some experiments to demonstrate whether an analysis that considers a hierarchical relationship between different units outperforms better than the analysis that fails to consider hierarchical relationships.

Before investigating if a hierarchical structure in analysis may help the performance of the analysis, we firstly show the existence of a hierarchical relationship for expressive timing by comparing the centroids of clustering of intra-phrase expressive timing. Examining the centroids of clusters with the doubled phrase length, we find that the centroids of clusters with doubled phrase length can be constructed from the centroids of clusters with the shorter phrase lengths.

This fact suggests there is a potential hierarchical relationship that exists for expressive timing.

To test the performance of a candidate hierarchical structure for analysis, we propose a method that converts expressive timing into the probability that every beat in a performance locates a boundary of music structure. Before introducing the method we proposed, we show the link between music structure and expressive timing. From the observation of Tempo Variegation Maps (TVMs) in section 4.1.3, there are certain phrases in which performers reach an agreement of the variations of expressive timing. We will examine the distribution of clusters of expressive timing in TVMs again to justify if the phrases at the end of a section in a performance may have a slowing down of tempo during their final beats. The potential links between expressive timing and music structure could be applied inversely: a deceleration in expressive timing may indicate a higher probability of locating a musical structure boundary at that point.

Thus, we propose a method that asserts the probability of locating a music structure boundary for every beat in the performance. This method uses windows of different sizes to detect the local minimum points in tempo variations. The methodology of using different sizes of windows is widely used in music information retrieval (such as [Yang et al., 2015]) for processing audio signals. Moreover, we can use different sets of windows to adapt different hierarchical structure in the analysis. As a result, evaluating the models corresponding to different sets of windows is equivalent to evaluating how much different hierarchical structures will help the analysis of expressive timing.

To evaluate the resulting models, we propose two tests. The first test uses the same principle of query likelihood. We use the music structure boundaries as the ground truth to query the resulting models in terms of how likely the ground truth is observed according to the resulting models. This test shows which structure helps to detect the boundaries of music structure according to expressive timing. The second test investigates which structure of analysis in expressive timing helps to show the similarity between performances from the same performer, or, in other words, same-performer renderings. According to

[Sapp, 2008], the 'same-performer rendering' should have similar expressive timing style. Thus, we design an experiment that compares the resulting models from same-performer renderings. We expect the models incorporating hierarchical structure to outperform the models without hierarchical structure to create more similar models from the same-performer renderings than from other performances. Thus, both tests will provide the evidence that a model for expressive timing should consider hierarchical structure in general.

The experiment of comparing same-performer rendering demands a larger database that contains a certain number of same-performer renderings. Furthermore, as the algorithm converting the original expressive timing into models does not require the identical length of phrases throughout a piece of performance, the lengths of phrases in the candidate pieces need not be consistent. Moreover, as the *Islamey* database does not contain any same-performer rendering, we will use the full Mazurka database that is used in [Sapp, 2008] for the experiments in this chapter. The full Mazurka database used in [Sapp, 2008] contains five Chopin Mazurkas: Op. 17/4, Op. 24/2, Op. 30/2, Op. 63/3 and Op. 68/3.

This chapter is organised in the following way. Firstly, we show evidence that expressive timings across different unit lengths are potentially linked. Then we demonstrate how to build the mathematical model to locate music structure boundaries. Next, we present how we evaluate the resulting models. Finally, we discuss the results, followed by a conclusion.

5.1 Hierarchical Relationship in Expressive Timing

In this section, we are going to show the existence of a hierarchical relationship in the clustering of expressive timing with different phrase lengths. The usage of clusters of expressive timing can be considered as a sequential model, for example, the sequential model we proposed in Chapter 4 and the string match in [Madsen and Widmer, 2006]. If we take the same problem from another point of view, we find that combining clusters of expressive timing sequentially

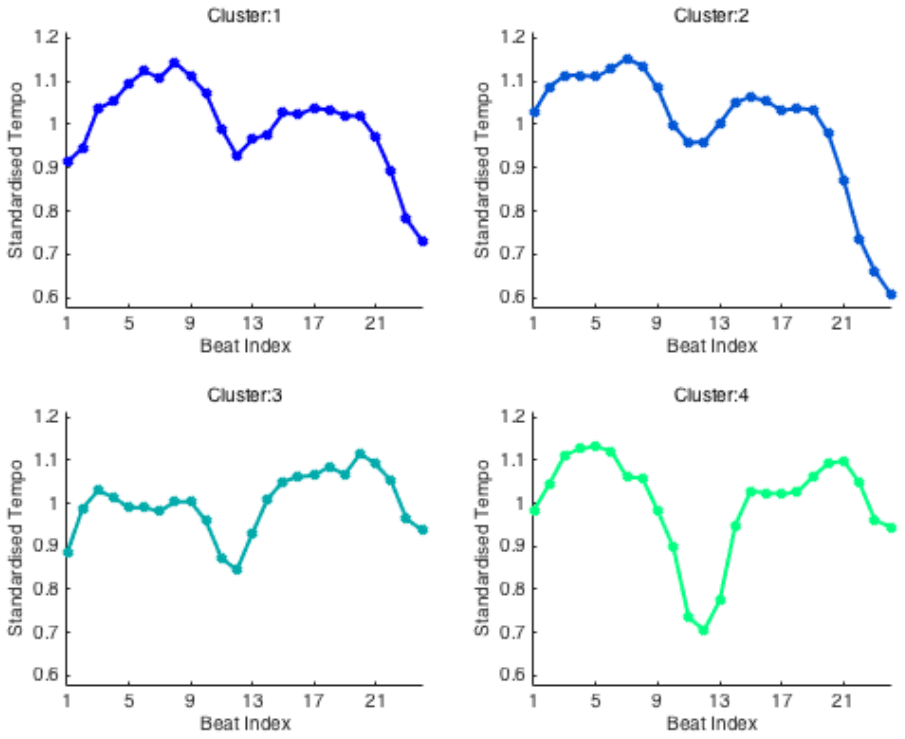


Figure 5.1: The centroids of clustering with 24-beat phrases in Chopin Mazurka Op.24/2.

is equivalent to clustering expressive timing within longer phrases. Now we repeat the experiment in Chapter 3 but double the length of phrase for analysis in order to investigate if there are potential hierarchical relationship between the centroids of clusters with different lengths of phrases engaged. By comparing the centroids of clusters with different lengths of phrase, we examine if the centroids of double-length phrase can be constructed from the centroids of shorter phrase.

In Figure 5.1, we have shown the centroids of the clustering of intra-phrase expressive timing of the Chopin Mazurka Op.24/2. The phrase length we engaged in Figure 5.1 is 24-beats, which is the doubled length of phrase provided by the Mazurka dataset. The centroids of clustering with 12-beats phrase, which is the length of phrase provided by the Mazurka dataset, are shown in the bottom of Figure 4.2.

By comparing the centroids of clusters with doubled length of phrases en-

gaged, we find that there are several centroids of longer phrases that are constructed by centroids of shorter phrases with different means of different parts. Here we present some observations:

- cluster 3 of 24-beat phrase is composite by cluster 2 and cluster 4 of 12-beat phrase.
- cluster 4 of 24-beat phrase is composite by cluster 8 and cluster 2 of 12-beat phrase.

These observations confirm that a potential hierarchical relationship exists in expressive timing as the centroids of clusters with longer length of phrases can be constructed from the centroids of clusters with shorter length of phrase. In this chapter, we are going to show that if the hierarchical structures are used for the analysis of expressive timing, the analysis may have a better performance.

The method we propose to compare different hierarchical structures for analysis of expressive timing is to use the candidate hierarchical structure to build up a model that asserts the probability of locating a boundary of music structure for every beat in a performance. The relationship between expressive timing and music structure boundaries is shown in Section 4.1. The proposed model will assert the probability of locating a boundary of music structure according to the values of tempo. We shall evaluate how well the models using different hierarchical structures predict music structure boundaries. In the next section, we will introduce the proposed model and how the proposed model is evaluated.

5.2 Methodologies

5.2.1 Model establishment

As we have demonstrated, performers slow down the tempo around the important boundaries of music structure. We propose a model that asserts the probability that every beat in the performance locates a music structure boundary. To make the proposed method capable to adapt different hierarchical structures, a multi-level window size is used.

Now we are going to introduce how the model asserts the probability of a beat in a performance being located at a boundary of music structure. According to [Todd, 1992], there is a potential hierarchical relationship in the tempo variations in performed music. This fact suggests that the global minimum points on tempo variations have a higher possibility to locate a boundary of music structure, whereas maxima of tempo variations are less likely to be located at a music structure boundary. We calculate the Root Mean Square (RMS) with different sizes of windows to indicate the minimum points of tempo variations at more global scales. Moreover, if the window size are very large, the RMS of a window may not be a local minimum point even if there is a global minimum point in the window. So we award a window that has a local minimum point by subtracting the standard deviation of tempo values in the window. To cover the local minimum points of tempo variations at various scales, the proposed method considers the minimum points at all levels of scales.

Now we give the mathematical representation of the proposed method. We use $\vec{\tau} = (\tau_0, \tau_1, \ldots, \tau_{n-1})$ to represent the value of tempo on every beat in a performance. The different size of windows is represented as $\mathbf{L} = \{l_0, l_1, \ldots, l_i, \ldots, l_L\}$ where $l_1 < l_2 < \ldots < l_L$. We use rectangle windows in this algorithm, which defines the window function $(W_{l_i}(\tau))$ of size l_i as:

$$W_{l_i}(\tau) = \begin{cases} 1 & \tau \in [0, l_i - 1] \\ 0 & \text{otherwise.} \end{cases} \tag{5.1}$$

For simplicity, we call the analysis with windows of size l_i as the analysis of level i. Level 0 is defined as the lowest level, and level L is defined as the top level (or the highest level).

We then calculate the Root of Mean Square (RMS) of tempo values in each window at different levels. If the windows size is l_i and the performance has n beats (from 0 to $n-1$), we should have $\dfrac{n}{l_i}$ windows. However, if n is not divisible by l_i, the size of the last window at level l will be shortened to fit the end of performance. If we use $\mathbf{e}_{l_i} = (e^1_{l_i}, e^2_{l_i}, \ldots, e^j_{l_i}, \ldots, e^k_{l_i})$ (where $k = \lfloor \frac{n}{l_i} \rfloor$) to represent the RMS values within each windows whose size is l_i. The sign of

$\lfloor \frac{n}{l_i} \rfloor$ represents the smallest integer that is larger than $\frac{n}{l_i}$. The element in $\mathbf{e}_{l_i}$ is defined as:

$$e_{l_i}^j = \begin{cases} \sqrt{\frac{\Sigma(\vec{\tau}^T W_{n-(j-1)l_i}(\tau - jl))^2}{n-(j-1)l_i}} - std(\vec{\tau}^T W_{n-(j-1)l_i}(\tau - jl)) & i = \lfloor \frac{n}{l} \rfloor \neq \frac{n}{l} \\ \sqrt{\frac{\Sigma(\vec{\tau}^T W_{l_i}(\tau - jl_i))^2}{l_i}} - std(\vec{\tau}^T W_{l_i}(\tau - jl_i)) & \text{otherwise} \end{cases} \tag{5.2}$$

where *std* represents the standard deviation. We need to point out that $\mathbf{e}_{l_i}$ has different lengths. The length of $\mathbf{e}_{l_i}$ is $\lfloor \frac{n}{l_i} \rfloor$.

Next, we introduce the way to convert a series of $\mathbf{e}$ into the probability of locating a music structure boundary for every beat in a performance. As the minimum points at different levels of $\mathbf{e}_{l_i}$ map the minimum points of tempo variations at different scales, the proposed methods consider all levels of $\mathbf{e}_{l_i}$ and award the minimum points of higher levels of $\mathbf{e}_{l_i}$. The higher levels of $\mathbf{e}_{l_i}$ find the minimum points of tempo variations at more global scales.

We now demonstrate how we calculate the probability of a beat in a performance being located at a boundary of music structure. Suppose there are n beats in a piece of performance; if we use $\mathbf{B}$ to represent all beats in a performance and b_k to represent each beat in a performance, we have $\mathbf{B} = \{b_1, b_2, \ldots, b_k, \ldots, b_n\}$, where the index of b represents the order of beats in the performance. If we use $b_k \in W_{l_i}^j$ to represent that beat k that belongs to at level i, then $p(\mathcal{B} \in (b_k \in W_{l_i}^j))$ represents the probability that a music structure boundary ($\mathcal{B}$) locates at beat k which is in the jth window at level i. The probability of a boundary of music structure in jth window at level i is proportional to the reciprocal of squared $\mathbf{e}_{l_i}$, e.g.

$$p(\mathcal{B} \in (b_k \in W_{l_i}^j)) = \frac{1/(\mathbf{e}_{l_i}^j)^2}{\sum_{x=1}^{\lfloor \frac{n}{l_i} \rfloor} (1/(\mathbf{e}_{l_i}^x)^2)}. \tag{5.3}$$

As the probability of locating a music structure boundary in windows at different levels are independent to each other, the probability of a beat b_k that locates a boundary of music structure ($p(\mathcal{B}) = b_k$) can be calculated by the

cumulative production of the probability of windows at each level that contain b_k, e.g.

$$p(\mathcal{B} = b_k) = \prod_{i=1}^{L} p(\mathcal{B} \in (b_k \in W_{l_i}^j)). \tag{5.4}$$

As we want to find out the probability of every beat in a performance that locates the music structure boundary, we must engage the lowest level in the set of window sizes to 1 (i.e. $l_0 = 1$), such that there is only 1 beat in the lowest level of windows.

5.2.2 Evaluation of resulting models

As the proposed algorithm enables us to build up models for a candidate performance according to different hierarchical structures, we want to select the best hierarchical structure that results in the most accurate model mapping music structure boundaries with expressive timing. To test the resulting models from a hierarchical structure, we propose two methods: the query likelihood test and the performance matching test. The query likelihood test shares the same principle of the query likelihood test in Chapter 4. We use the phrase boundary provided in the Mazurka database to query the probability of observing the phrase boundaries provided. We would expect a better model should have a better chance to observe the boundaries of music structure provided. Specifically in this experiment, if a hierarchical structure performs well for analysing expressive timing, the resulting model should give a higher likelihood for observing the music structure boundaries provided by our database.

A more objective test is matching the performances from the same performer. As discussed in [Sapp, 2008], the performances from the performers can be expected as similar in expressive timing. So we compare the similarity of the models that are resulted by the same-performer renderings. We expect the models for same-performer rendering are more similar than other resulting models, especially when some certain hierarchical structures are used. Next, we are going to give more details about how both methods of evaluation are implemented.

Query likelihood

In this section, we will introduce one of the way to validate the proposed mathematical models: the query likelihood test. For the same piece of performance, we can use different hierarchical structures for the model establishment process. We query the music structure boundaries provided by the Mazurka database to the resulting models with different hierarchical structures. The result of this experiment reveals which structure of analysis helps locate the boundaries of music structure according to expressive timing.

Next, we are going to introduce how we calculate the log-query likelihood. Assume that there are n beats $\mathbf{B} = \{b_1, b_2, \ldots, b_n\}$ and m music structure boundaries $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_m\}$ in a piece of performance. Each music structure boundary $\mathcal{B}_j$ is located on kth beat of the performance ($\mathcal{B}_j = b_k$). The log-query likelihood (q) is

$$q = \log\left(\prod_{i=1}^{m} p(B_i = b_k)\right) = \sum_{i=1}^{m} \log p(B_i = b_k) \tag{5.5}$$

where $k \in [1, n], k \in N^+$.

A higher log-query likelihood indicate a better performance, as the boundaries of music structure provided in the database are likely to be observed. We compare the log-query likelihood of the resulting models that are derived by different hierarchical structures.

Similarity of same-performer renderings

In [Sapp, 2008], the author used a correlation-based similarity measurement to compare the performances from the same performer, or say, same-performer renderings. According to this work, same-performer renderings are more similar to each other than the different-performer renderings. In this experiment, we investigate which structure of analysis helps show the similarity between the same-performer renderings.

To show the similarity between same-performer renderings, we compare the resulting model we proposed from all renderings including the same-performer

renderings and the different-performer renderings first. For a particular hierarchical structure, the similarity of resulting models are compared by Kullback-Leibler divergence [Murphy, 2012, p. 58]. The different hierarchical structure used may lead to different mean and variance in terms of KL divergence for the model similarity between all renderings in the database. To avoid the possible effects caused by the difference of mean and variance of model similarity, we order the similarity between one rendering (for simplicity, we call this rendering 'target rendering') and all other renderings. If we use all renderings as the target rendering once iteratively, we have the average rank of the rendering similarity between all same-performer renderings. A lower (smaller) average rank of the rendering similarity for the same-performer renderings indicates a better performance of resulting models, as the models show the similarity of same-performer renderings.

To compare the performance of analysis with different hierarchical structures, we repeat the experiment with candidate hierarchical structures. We compare the average rank of the rendering similarity for the same-performer renderings with candidate hierarchical structures. The candidate structure that results in a lowest average rank of the rendering similarity for the same-performer renderings is better than the other candidate hierarchical structures for analysis.

Although the average rank of similarity of the same-performer renderings is a straightforward measurement, the lowest possible average rank of rendering similarity for the same-performer renderings is different according to the number of same-performer renderings from a particular performer. For example, the lowest possible average rank of rendering similarity for two same-performer rendering is 1 (excluding the rendering being compared), but the lowest possible average rank of rendering similarity for three same-performer rendering is 1.5 (excluding the rendering being compared). This is because if there are other three same-performer renderings in the database (excluding the rendering being compared), the lowest possible rank of rendering similarity for the first same-performer rendering is 1, but the lowest possible rank of rendering similarity for the second same-performer rendering is 2. If there are n same-performer

renderings in the database whose rank of rendering similarity to another same-performer rendering is $\{R_1, R_2, \ldots, R_i, \ldots R_n\}$, we correct the bias of lowest possible rank of rendering similarity by calculating the average rank of rendering similarity ($\mathcal{R}$) as:

$$\mathcal{R} = \begin{cases} R_1 & n = 1 \\ \frac{\sum_{i=1}^{n} R_i - \sum_{i=1}^{n} i}{n} & n > 1, n \in N^+. \end{cases} \tag{5.6}$$

5.2.3 Candidate hierarchical structures

We propose several hierarchical structures ($\mathbf{L}_0$ to $\mathbf{L}_7$) for comparison. As we discussed in 5.2.1, the symbol $\mathbf{L}$ represents a series of window sizes for analysis. To represent each beat in the performance, the lowest level (l_0) in $\mathbf{L}$ is always 1. As a result, the lowest adjustable level in the candidate hierarchical structures for analysis if l_1.

In the selected Chopin Mazurkas, there are three beats in each bar. So in most cases (except only two cases for comparison purposes), the second lowest level (l_1) of $\mathbf{L}$ is set to be 3 ($l_1 = 3$). We want to investigate the effects of the multiple levels used. We set higher levels in the candidate hierarchical structure to 2 to make the window size at top-level expand at a proper speed. The largest window size at top-level in the candidate hierarchical structures is 48 beats, which is about at least twice longer than the length of phrases provided in the Mazurka database. There are two special cases in the candidate hierarchical structures by setting different values to l_1 for comparison purposes. The first special hierarchical structure has $l_1 = 6$, which is twice longer than the length of a bar. We want to investigate the effects that l_1 covers a longer section of a performance. The second special hierarchical structure engaged is $l_1 = 4$, which even breaks the bar at certain point. We wish to investigate if it is important to make l_1 to be identical to the length of bar, or, at least do not break a bar.

The candidate hierarchical structures are listed in Table 5.1. For simplicity, we use symbols (M_1 to M_7) to represent the models resulting from candidate hierarchical structures L_1 to L_7.

Candidate Hierarchical Structures
$L_0 = \{1\}$
$L_1 = \{1, 3\}$
$L_2 = \{1, 3, 6\}$
$L_3 = \{1, 3, 6, 12\}$
$L_4 = \{1, 3, 6, 12, 24\}$
$L_5 = \{1, 3, 6, 12, 24, 48\}$
$L_6 = \{1, 6, 12, 24\}$
$L_7 = \{1, 4, 12, 24\}$

Table 5.1: The candidate hierarchical structures for analysis in this experiment.

5.3 Results

5.3.1 Query likelihood test

According to (5.5), we calculate the query likelihood for five piece of Mazurkas in the database. The query likelihood is shown in Table 5.2, where a more negative number means worse performance of models.

From the results, we notice that on average $\mathbf{L}_1$ and $\mathbf{L}_2$, that have fewer levels in hierarchical structures have a more negative query likelihood than the non-hierarchical structure $\mathbf{L}_0$. The more hierarchical structures ($\mathbf{L}_3$ to $\mathbf{L}_7$) have better results. This fact suggests that the hierarchical structures for analysis of expressive timing should have a sufficient number of layers for the performance of the analysis.

Moreover, when comparing the query likelihood results of different pieces of Mazurkas, we noticed that the candidate hierarchical structures of analysis that give the best performance may differ from each other. For Op.63/3, Op.68/3 and on average, the top level in the best performing hierarchical structure covers 24 beats of the performance. Moreover, the resulting query likelihood for other pieces with a hierarchical structure whose top level covers 24 beats is reasonably well (only less than 1 per cent worse than the best performance in terms of query

$\mathbf{L}_i$ \ Opus	Op.17/4	Op.24/2	Op.30/2	Op.63/3	Op.68/3	Avg.
$\mathbf{L}_0$	-7.86	-8.19	-6.31	-7.06	-6.86	-7.26
$\mathbf{L}_1$	-7.83	*-8.28*	**-6.19**	-7.04	*-7.01*	-7.27
$\mathbf{L}_2$	*-8.36*	-8.03	*-6.70*	*-7.28*	-6.95	*-7.47*
$\mathbf{L}_3$	-8.02	**-7.91**	-6.30	-7.18	-6.51	-7.18
$\mathbf{L}_4$	-7.69	-7.96	-6.24	-7.08	**-6.50**	-7.09
$\mathbf{L}_5$	**-7.65**	-8.02	-6.25	-7.10	-6.67	-7.14
$\mathbf{L}_6$	-7.66	-8.00	-6.25	-7.01	-6.56	-7.10
$\mathbf{L}_7$	-7.67	-7.98	-6.21	**-7.00**	-6.52	**-7.07**

Table 5.2: The average query log likelihood for resulting models given hierarchical structure $\mathbf{L}$. The bold number shows the best result for a piece of Mazurka and the italic number shows the worst result for a piece of Mazurka.

likelihood). Surprisingly, the candidate structure resulting best query likelihood in Op.30/2 is $\mathbf{L}_1$, which is the worst structure for analysis for Op.24/2 and Op.68/3. The hierarchical structure resulting the most negative query likelihood for other candidate pieces is $\mathbf{L}_3$.

This result suggests that, for the query likelihood test, the hierarchical structure resulting best query likelihood should cover a proper part of the performance. However, if the coverage of hierarchical structure at top level is too large, the query likelihood may be more negative. Moreover, the proper coverage for the top level of the hierarchical structure for analysis may differ from piece to piece.

5.3.2 Similarity between same-performer renderings

We now present the average rank of rendering similarity between same-performer renderings in Table 5.3. In this table, a lower average rank of rendering similarity between same-performer renderings suggests a better performance of resulting models with the candidate hierarchical structure for analysis.

$\mathbf{L}_i$ \ Opus	Op.17/4	Op.24/2	Op.30/2	Op.63/3	Op.68/3	Avg.
$\mathbf{L}_0$	*5.93*	**2.20**	**2.22**	3.56	4.33	*3.65*
$\mathbf{L}_1$	2.17	*3.20*	2.33	*4.90*	5.14	3.55
$\mathbf{L}_2$	2.17	**2.20**	*3.78*	**2.51**	*5.83*	3.30
$\mathbf{L}_3$	2.20	**2.20**	2.89	2.53	5.28	3.02
$\mathbf{L}_4$	2.17	**2.20**	2.50	2.70	5.00	2.91
$\mathbf{L}_5$	2.33	**2.20**	2.50	2.77	4.17	2.77
$\mathbf{L}_6$	**2.13**	**2.20**	2.89	2.79	**4.08**	2.82
$\mathbf{L}_7$	2.17	**2.20**	2.39	2.86	3.94	**2.71**

Table 5.3: The average rank of rendering similarity between same-performer renderings with different hierarchical structures ($\mathbf{L}$) in the analysis. The bold number shows the best result for a piece of Mazurka and the italic number shows the worst result for a piece of Mazurka. For Op.24/2 several statistics are in bold due to a tie.

We first observe the average ranks across different pieces of mazurkas. The results are shown in the first column on the right-hand side in Table 5.3. In general, comparing $\mathbf{L}_0$ to $\mathbf{L}_1$, the average rank of rendering similarity between same-performer renderings becomes smaller with the increase of coverage at the top level in the candidate hierarchical structure. Moreover, if the candidate hierarchical structure has the same coverage at the top level ($\mathbf{L}_4$, $\mathbf{L}_6$ and $\mathbf{L}_7$), a larger l_1 helps to show the similarity between same-performer rendering (comparing $\mathbf{L}_6$ and $\mathbf{L}_7$ to $\mathbf{L}_4$). However, the effects of larger coverage of l_1 requires further investigation as $\mathbf{L}_6$ has a larger l_1 than $\mathbf{L}_7$, but the average rank of rendering similarity between same-performer renderings with $\mathbf{L}_6$ is higher than $\mathbf{L}_7$; in other words, $\mathbf{L}_7$ is better than $\mathbf{L}_6$ for finding similarity between same-performer renderings.

If we observe the results in Table 5.3 piece by piece, the case is much more complicated. This fact suggests that the most suitable hierarchical structure

to show the similarity between same-performer renderings depends on the particular Mazurka being modelled. Taking Op.24/2 as an example, the average rank of rendering similarity between same-performer renderings does not change with the hierarchical structure used in most cases, which suggests the different hierarchical structures do not help to show the similarity of same-performer renderings of Op.24/2. For Op.30/2 the non-hierarchical analysis outperforms all hierarchical clusterings. The best hierarchical structure to show the same-rendering similarity for Mazurka Op.63/3 is $\mathbf{L}_3$, which only contains three levels. The Op.17/4 and Op.68/3 show a clear preference for more hierarchical structures with a larger l_1 level ($\mathbf{L}_6$). This fact suggests that, in order to show the similarity of same-performer renderings, the most suitable hierarchical structure differs from piece to piece.

5.4 Discussion

In this chapter, we proposed an method that asserts the probability of a beat in a performance being located at a boundary of music structure according to expressive timing information. This method is applicable to different hierarchical structures. We design two experiments that compare the effects of candidate hierarchical structures.

The first experiment tests which hierarchical structure helps to detect the boundaries of music structure. We tested the probability that the boundaries of music structure provided in the Mazurka database are observed according to the resulting models. This experiment is called query likelihood test. According to the results of the query likelihood test in Table 5.2, we noticed that, on average, the three best hierarchical structures ($\mathbf{L}_4$, $\mathbf{L}_6$ and $\mathbf{L}_7$) for detecting boundaries of music structure share the same coverage at the top level of the hierarchical structure (24 beats). This fact suggests that, in general, a hierarchical structure whose top level covers an appropriate length of performance may help to detect the boundaries of music structures according to expressive timing. In other words, a hierarchical structure is necessary to detect the boundaries of music

structure according to expressive timing.

On the other hand, the hierarchical structures that outperform the other hierarchical structures in Op.17/4, Op.24/2, Op.30/2, Op.63/3 and Op.68/3 are $\mathbf{L}_5$, $\mathbf{L}_3$, $\mathbf{L}_1$, $\mathbf{L}_7$ and $\mathbf{L}_4$ respectively. This fact suggests that, for different candidate pieces, the most suitable hierarchical structure that helps to detect the boundaries of music differs from piece to piece.

Moreover, it is interesting to point out that, on average, the best hierarchical structure in Table 5.2 is $\mathbf{L}_7$, which is the only candidate hierarchical structure that breaks bars in level 1. This fact suggests that the hierarchical structure that helps detecting boundaries of music structure may break the bars in the performances at certain levels.

The second experiment investigates what hierarchical structure used in the analysis of expressive timing helps to show the similarity between the same-performer renderings. We used the average rank of rendering similarity between the same-performer renderings to assess the performance of candidate hierarchical structures. The results in Table 5.3 show that, in general, the candidate hierarchical structures $\mathbf{L}_4$, $\mathbf{L}_5$, $\mathbf{L}_6$ and $\mathbf{L}_7$ have better performance than those hierarchical structures that are less hierarchical. As a result, the hierarchical structure that helps to show the similarity between same-performer renderings should have a top level that covers appropriate length of performance. In other words, the hierarchical structures help to show the similarity between same-performer renderings.

If we examine the performance of candidate hierarchical structures piece by piece, we notice that for different pieces the best performed hierarchical structure is also different. In fact, if we consider each individual piece in the database, the hierarchical structure may not help to show the similarity between same-performer renderings for all pieces according to Table 5.3. For example, the average rank of rendering similarity between same-performer renderings does not change with most hierarchical structures used in the analysis for Op.24/2. For Op.30/2, the best structure that leads to best (lowest) average rank of rendering similarity between same-performer renderings is a non-hierarchical

structure. This fact suggests, similarly to the query likelihood test, that the most suitable hierarchical structures for showing the similarity between same-performer renderings differ from piece to piece. Another conclusion we can draw from Table 5.3, again similar to the conclusion in query likelihood, is that $\mathbf{L}_7$ is the most suitable hierarchical structure to show the similarity between same-performer renderings on average. As $\mathbf{L}_7$ is the only candidate hierarchical structure that breaks bars in the performances, the hierarchical structure that is suitable to show the similarity between same-performer renderings may break the bars in performances at certain levels.

Summarising the results in both test, the hierarchical structure in general helps the analysis of expressive timing on average. For different pieces of performances, the most suitable hierarchical structures for analysis are different. Moreover, the most suitable hierarchical structure for analysis of expressive timing may break the bars in performances.

5.5 Conclusions

In this chapter, we tested different structures of analysis for expressive timing. We proposed a model that asserts the probability of every beat in a performance that locates a boundary of music structure. With the proposed model, we test which hierarchical structure is helpful for detecting the boundaries of music structure and which hierarchical structure is helpful for showing the similarity between the same-performer renderings.

In Chapter 3, we demonstrated that intra-phrase tempo variations can be modelled by the Gaussian Mixture Model (GMM). The phrases are defined by either the composers or the information in the dataset. We repeated the experiment with doubled phrase length and found the centroids of clusters with the doubled phrase length can be constructed from the centroids of clusters with shorter length of phrase. As a result, we found that there is a potential hierarchical relationship in expressive timing.

In Chapter 4, we introduced the Tempo Variegation Map (TVM) for visu-

alisation. From the observations of TVMs, we noticed that the deceleration of expressive timing can be potentially used to locate the boundaries of music structure. So the expressive timing can be used to detect the boundaries of music structure.

The model we proposed in this chapter converts expressive timing into the probability of every beat in the performance that locates a boundary of music structure. This method is capable of adopting a hierarchical structure for analysis during the modelling process. We proposed a few different hierarchical structures as inputs, then evaluated the resulting models by two different experiments.

To evaluate the model performances with different hierarchical structures, we performed two model selection tests. The first one is using the boundaries of music structure provided in the Mazurka dataset to query the probability of observing such sets of music structure boundaries according to the resulting models. The second test is investigating how well the resulting models show the similarity between same-performer renderings. The measurement we selected for showing similarity is the average rank of rendering similarity between the same-performer renderings.

We compared the performance of several hierarchical structures in both tests including a non-hierarchical structure and several hierarchical structures that keep the bars in the performances. Moreover, for comparison purposes, there is also a hierarchical structure that matches bar starts and ends in the performance but covers a larger area in the lower levels, and a hierarchical structure that breaks bars in the performances. In total, seven candidate hierarchical structures are tested.

Based on the results, we conclude that having a hierarchical structure in expressive timing analysis is important. In general the top level of a hierarchical structure needs to cover a sufficiently large number of beats. Moreover, the hierarchical structure used in the analysis does not have to synchronise with bar starts and ends. Finally, there is no one hierarchical structure that best models all performances in our database.

Chapter 6

Conclusions

6.1 Summary

In this thesis, we demonstrated that it is possible to use model selection tests to analyse expressiveness in performed piano music, especially to analyse expressive timing. With this approach, we investigate three factors related to expressive timing in performed music: clusters of expressive timing, factors that are used to determine expressiveness and the importance of hierarchical structures in analyses of expressive timing. However, it is possible that model selection methods could be used for various aspects of expressive timing, such as synchronisation between hands [Goebl et al., 2010] and changes in expressive styles over time [Flossmann et al., 2009]. The model selection method may also be used to analyse other behaviour in performed music, such as dynamics [Repp, 1999a] and pedal timing [Repp, 1996]. Next, we will review some of the specific experiments we performed.

We first investigated if the expressive timing within a phrase can be clustered. Then we examined how the position and changes of expressive timing in the previous phrase impact the changes of expressive timing in a particular phrase. Finally, we investigated if the hierarchical structure helps the analysis of expressive timing.

In Chapter 3, we first demonstrated that expressive timing within a phrase

can be clustered. We considered several candidate models, including both non-clustered and clustered models. Moreover, we compared the performance of candidate models with different settings, in order to investigate how the data of expressive timing can be better fitted. We tried two methods to evaluate the proposed models: cross-validation tests and model selection criteria. In cross-validation, a model is tested by evaluating how well a part of the data is predicted by the remaining data; whereas the use of model selection criteria involves utilising selected criteria that penalise the model performance according to the model complexity. Both evaluations indicated that the clustered models outperform the non-clustered models. Moreover, by comparing different settings for the candidate models, we found that the best model for clustering expressive timing within a phrase is the Gaussian Mixture Model with independent full covariance matrices and with data standardised by the Mean Regulation. As a summary, expressive timing can be clustered by a normal Gaussian Mixture Model but the number of Gaussian components in the model varies according to the different pieces.

In Chapter 4, we then explored how the use of clusters of expressive timing is impacted by the position of the phrase and the cluster of expressive timing used in the previous phrase. We proposed four different Bayesian graphical models that set up different dependencies for the expressive timing in a particular phrase. The proposed models included: 1) an independent model, which considered both the expressive timing in the previous phrase and the position of the phrase in a score as independent factors for expressive timing, 2) a positional model, which considered only the position of the phrase in a score as the decisive factor for expressive timing in a phrase, 3) a sequential model, which considers only the expressive timing in the previous phrase as the decisive factor for expressive timing in a phrase and 4) a joint model, which considers both the expressive timing in the previous phrase and the position of the phrase as the factor for expressive timing in a phrase jointly. We proposed a novel model selection criterion, the Cross-Entropy Ratio (CER), which uses the principle of information theory to measure the performance of the proposed models in

a cross-validation test. The results showed that the joint model outperforms the other models but is less robust due to complexity. The second best model was the sequential model but this was more data-size robust. These conclusions tell us both the position of the phrases and the cluster of expressive timing used in the previous phrase impact on the cluster of expressive timing used in a phrase but that the effects of the position of the phrases are based on a joint consideration of the cluster of expressive timing used in the previous phrase.

In Chapter 5, we investigated whether hierarchical structure is necessary for the analysis of expressive timing. We introduced a way to convert the expressive timing into a probability that locates a boundary of music structure on every beat of a performance. A hierarchical structure can be used as an input into the proposed method. There are several candidate hierarchical structures that can be used and that result in different models. To evaluate the resulting probabilistic models according to the different candidate hierarchical structures, we designed two experiments. The first experiment compared the probability of observing the boundaries of music structure provided by the Mazurka database according to the resulting data. The second experiment investigated how well the resulting models show similarities in the same-performer rendering. From the results of both experiments, we can conclude that on average, a hierarchical structure may help the performance of the analysis of expressive timing. However, the best structure of analysis for expressive timing for each individual piece of performance may vary piece to piece.

Through the experiments performed in this thesis, we have shown that model selection tests can be used to analyse expressiveness in classical music, especially expressive timing in classical music. Model selection tests can potentially benefit further investigations of expressive timing and benefit the analysis in other aspects of expressiveness.

6.2 Future Works

In this thesis, we have described the experiments using model selection tests to demonstrate a number of basic principles associated with analyses of expressive timing. However, other parameters in these experiments are still unclear. In this section, we will discuss some specific works that would be interesting to investigate.

In Chapter 3, we showed that a Gaussian Mixture Model (GMM) can be used to fit the data of expressive timing within a phrase. The method we used for training the GMMs was the Expectation Maximum (EM). One of the most important aspect for EM is the initial values of the parameters in the model [Karlis and Xekalaki, 2002]. In Chapter 3, we chose a random sample in the training dataset and repeated the training process several times and chose the resulting model with the performance best fitted to the data of expressive timing. However, there are many other possibilities of using different initial points. As a result, we recommend a further investigation into the impacts of the initial values of the training process on the performance of the resulting models, as a good initial value in EM may help the training process of GMM become more efficient and lead to a better performance of the resulting models.

Moreover, the phrase lengths we selected for the testing pieces are not demonstrated in Chapter 3. As a result, one possible future work could analyse decisions regarding phrase length, which can be affected by the absolute playing time of a phrase. However, the best phrase length in an analysis is unlikely to be the same across different pieces of music. The proposed algorithm in Chapter 3 should be further developed such that model selection tests can be performed with different pieces of music. One possible solution to clusters of expressive timing across different pieces of music is to use normalised timing scales rather than a beat index.

In Chapter 4, we only considered the view that the position of a phrase may affect the use of clusters of expressive timing. However, the use of clusters of expressive timing may be impacted by several factors related to the music structure and which could affect the use of clusters of expressive timing. For

example, the melody of the phrase may be an important factor to be considered in deciding which cluster of expressive timing to use. The characterisation of the melody of a phrase may be represented by Implication-Realization (IR) theory [Narmour, 1995]. IR theory describes the audience expectation of melody. The core idea of IR is to use a sequence of two notes to predict the expectation of the real third notes in the score. An out-of-expectation note in a melody would naturally attract more attention from the audience. As a result, if we use IR theory to represent the features of melody, we then propose a more complicated Bayesian graphical model to reveal the potential relationship between the melody of a phrase and the cluster of expressive timing used for that phrase.

In some cases, it is worth investigating whether the music structure may also impact the use of clusters of expressive timing. A commonly used way to analyse music structure is to use the Generative Theory of Tonal Music (GTTM) [Lerdahl and Jackendoff, 1983]. GTTM covers the structure of tonal music with grouping structure, metrical structure, time-span reduction and prolongational reduction. A structural tree can be obtained, which shows a clearer image of how performances are structured. As a result, we may also consider if any particular factor in the analysis of a music structure impacts on the decision of the clusters of expressive timing for a phrase. To investigate the potential impacts, we may build a more complicated graphical model for testing.

Moreover, the GTTM theory can also be used for providing a more suitable hierarchical structure for the analysis of expressive timing. As we discussed in Chapter 5, the most suitable hierarchical structure for the analysis of expressive timing varies from piece to piece. It would be interesting to investigate how the most suitable hierarchical structures for different pieces of music are decided.

Besides music structure, there are several factors may affect expressive timing including melodies, harmonics and metrical patterns as we discussed in Section . To investigate the effects of these musical parameters, a more complicated graphical model should be considered. However, the structure and parameter learning for a more complicated graphical model demands more samples for training thus a larger database is required.

In Chapter 5, we demonstrated that the same-performer renderings show a very strong similarity. So it is worthy to investigate how the expressive timing can be used to match the different renderings from the same performer.

As the data used in this thesis are limited to two databases, applying the proposed methods with other pieces of performance would help to investigate if these methods can be generalised. Moreover, besides the expressive timing, there are several other factors in performed music that can be investigated, such as the dynamics. In fact, in computational musicology, the methods applied to expressive timing can also be applied to the dynamics in performed music with minor corrections, such as in [Repp, 1998], [Repp, 1999a] and [Repp, 1999b]. As a result, investigations into applying model selection methods to the research of dynamics in performed music is also an interesting future direction.

In this thesis, we have shown that model selection tests can be applied to the analysis of expressive timing in performed music. This thesis adds to the many works on the combination of computational musicology and mathematics and the combination of computational musicology and machine learning. Such combinations have contributed much to our understanding of how performers play music. We hope the combination of computational musicology and model selection tests may help the investigation of the behaviours of performers in the future.

Appendix A

Performances in *Islamey* database

as available in iTunes™ on 11/30/2015

Abdel Rahmanel Bacha

Russian Virtuoso Piano Works

Octavia Records Inc., 2006

Adam Aleksander

Balakirev, Chopin, Beethoven, Szymanowski, Liszt, Rachmaninov & Debussy: Works for Piano (Live)

Adam Aleksander, 2012

Alfred Brendel

Strauss li: Egyptischer - Saint-Saens, C.: Africa - Balakirev, M.A.: Islamey (Egyptian Splendour - The Mystery of Egypt in Classical Music)

Gift of Music, 2008

Andrei Gavrilov

Prokofiev/Tchaikovsky: Piano Concertos etc.

2005 Andrei Gaavrilov, London Symphony Orchestra, Philharmonia Orchestra, Riccardo Muti & Sir Simon Rattle

---

Arto Satukangas

Glazunov: Piano Sonata No. 1; Three Etudes

Finlandia Records, 1995

Aya Nagatomi

Islamey

Nipponophone, 2010

Barbara Nissman

Journeys of the Soul: From Bach to Balakirev - Barbara Nissman, Piano

Three Oranges Recordings, 2014

Boris Berezovsky

Berezovsky plays Mussorgsky, Rachmaninov, Liadov, Medtner

Teldec Classics International GmbH, 1994

Eileen Joyce

Rachmaninoff: Piano Concerto No. 2 in C Minor - Balakirev: Islamey

Past Classics, 2011

Emil Gilels

Emil Gilels: Recital in Florence (1951)

Music and Arts Programs of America, 2012

György Cziffra

Tchaikovsky: Piano Concerto No. 1, Op, 23 - Balakirev: Islamey (Mono Version)

BNF Collection, 2013

Idil Biret

Idil Ciret Archive Edition, Vol. 11

Idil Biret Archive, 2011

Janö Jandó

Mussorgsky: Pictures At Exhibition

Naxo, 1991

Jie Chen

Van Cliburn International Piano Competition Preliminary Round - Jie Chen

Van Cliburn Foundation, 1995

Jong-Gyung Park

Queen Elisabeth Competition: Piano 2003

Queen Elisabeth Competition, 2003

Lang Lang

Lang Lang - Complete Recordings 2000-2009

Deutsche Grammophon GmbH, Berlin, 2012

Michael Lewin

Piano Recital: Lewin, Michael - Balakirev M.A. - Scriabin, A. - Glazunov, A.K.

Centaur Records, Inc., 1992

Michele Campanella

Musorgskij - Balakirev: Pictures at an Exhibition, Ricordi d'infanzia, La cucitrice, Berceuse, Islamey Fantasy

P&P Classica, 2005

Mikhail Kollontay

Balakirev: Piano Works

Russian Season, 1995

Alvaro M. Rocha

The Beyond Piano Project

http://www.alvaromrocha.com, 2013

Olga Kern

Rachmaninov: Sonata No.2 - Balakirev: Islamey

Harmonia Mundi USA, 2006

Philip Edward Fisher

Piano Works by 'The Mighty Handful'

CHANDOS, 2011

Roger Wright

At the River

Roger Wright, 2011

Rorianne Scherade

Balakirev, M.A.: Piano Music

Centaur Records, Inc., 1995

Saito Kazuya

International Neo Classical Competition Prize Winners Concert Live 2010

(Ginza International Music Gestival)

Office ENZO & Florestan, 2010

Appendix B

Colouring Schemes for Tempo Variegation Maps (TVMs)

In Section 4.1.2, we presented one of the resulting centroids of clustered expressive timing in Figure 4.1. We then presented two different TVMs with colouring schemes focused on different criteria of the centroids of the resulting clustered expressive timing. In this section, we provide more details about the colouring schemes used. The descriptions in this section require a basic knowledge of colour space, especially RGB colour space. Please refer to [Poynton, 2003] if necessary.

To begin, we present Figure 4.1 again as Figure B.1 to present the centroids of resulting clustered expressive timing discussed in Section 4.1.2. Then, we will discuss how the centroids are coloured according to different criteria.

B.1 Shapes of centroids

The first criterion we selected is the shapes of the centroids. In Figure B.1, we find three types of clusters: single-peaked arcs (cluster 1 through cluster 4;

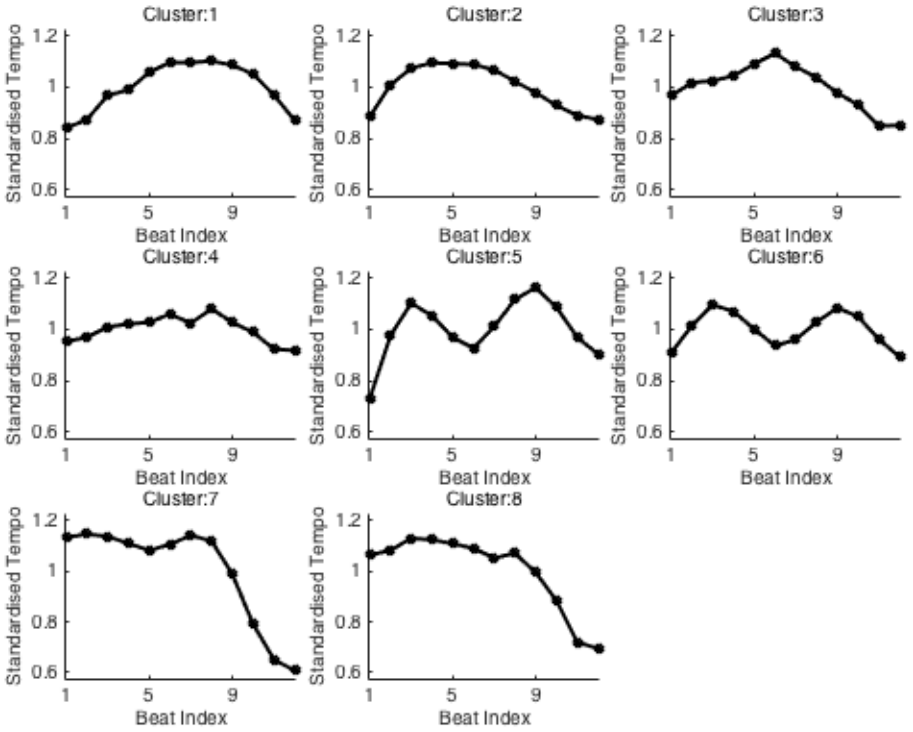


Figure B.1: Centroids of the clusters of expressive timing in Chopin Mazurka Op.24/2.

cluster 4 is a nearly single-peaked arc), double-peaked arcs (clusters 5 and 6) and decelerate arcs (clusters 7 and 8).

If we use red, green and blue to represent cluster 1, cluster 5 and cluster 7, respectively, and all other centroids are represented by intermediate colours, we can obtain a map that visualises the use of clusters of expressive timing throughout performances across different performers, as shown in Figure B.2. We call this type of diagram a Tempo Variegation Map (TVM). Based on certain observations of the resulting TVM, we can hypothesise which factor may affect decisions on clusters of expressive timing.

To enable an easier interpretation of the TVM, we also give the specific values of the colour codes used for each cluster in Table B.2. All the colour codes are represented in RGB colour space [Poynton, 2003] with a maximum value of 1. To make the colours used in the scheme more distinguishable, we

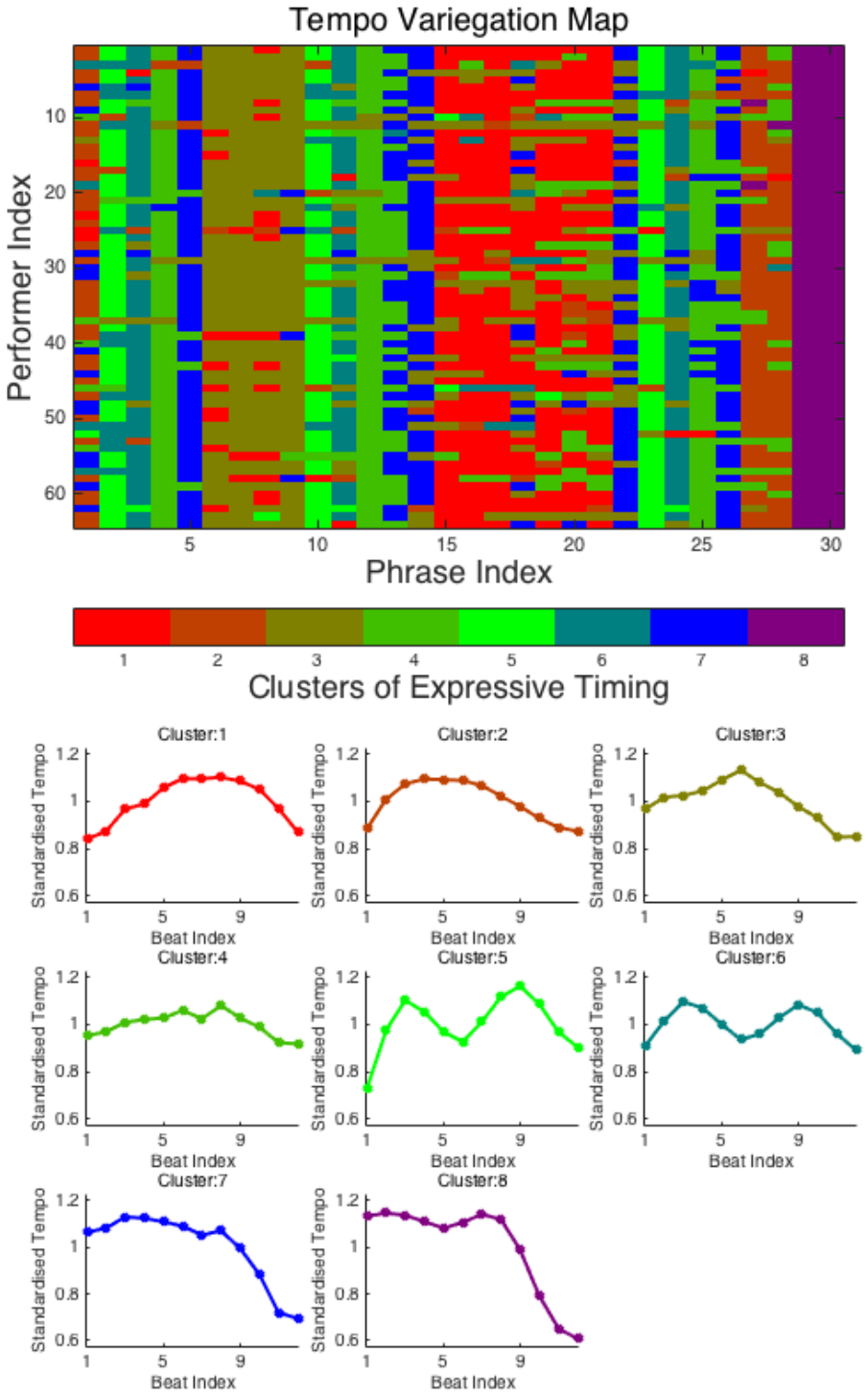


Figure B.2: Tempo Variegation Map (TVM) and the colours of the clusters of expressive timing. This is a repeated version of Figure 4.2.

do not normalise the brightness of the colours selected. However, different observations and hypotheses could be made if the brightness of the colours used is the same.

Cluster	R	G	B
1	1.00	0.00	0.00
2	0.75	0.25	0.00
3	0.50	0.50	0.00
4	0.25	0.75	0.00
5	0.00	1.00	0.00
6	0.00	0.50	0.50
7	0.00	0.00	1.00
8	0.50	0.00	0.50

Table B.1: The colour codes used for the clusters shown in Figure 4.1. Colours are represented by RGB colour space. The index of clusters are shown in Figure B.1.

B.2 According to the acceleration rate of centroids

Besides the order of clusters presented in Figure B.1, we can order the resulting centroids with different criteria such as the accelerate rate of centroids. For simplicity, we want to make the centroids become a more regular shape by performing a regression. As Todd [Todd, 1992] asserted, parabolic curves (2nd order polynomials) can be used to approximate tempo variations within a phrase. We thus regress the tempo variations to parabolic curves. The cluster of expressive timing is ordered according to the changes in the regressed parabolic curves rather than according to the unprocessed centroid of the cluster of expressive timing. We appoint the cluster whose regressed parabolic curve has the highest acceleration of tempo rate within a phrase as cluster 1, and the cluster whose

regressed parabolic curve has the greatest deceleration of tempo rate within a phrase as cluster A.

As a result, the numerical representation of the clusters of expressive timing show not only which clusters are used by performers but also the tempo changes within a phrase. A cluster with a smaller index represents a cluster of expressive timing whose tempo accelerates within a phrase. A cluster with a larger index represents a cluster of expressive timing whose tempo rate decelerates within a phrase.

We use RGB colour space [Poynton, 2003] to represent the colours in our colour code. We use blue ($(R, G, B) = (0, 0, 1)$) to represent cluster 1. Then we increase the Green component in RGB and we simultaneously decrease the Blue component in RGB simultaneously. The cluster A is represented as a green colour ($(R, G, B) = (0, 1, 0)$). The intermediate clusters are represented by the colours at even intervals of the intermediate values between the blue colour ($(R, G, B) = (0, 0, 1)$) and the green colour ($(R, G, B) = (0, 1, 0)$) in the RGB space. The specific colours used in Figure B.3 is presented in the format of RGB in Table B.2.

Cluster	R	G	B
1	0.00	0.00	1.00
5	0.00	0.14	0.85
2	0.00	0.28	0.71
4	0.00	0.42	0.57
3	0.00	0.57	0.42
6	0.00	0.71	0.28
7	0.00	0.85	0.14
8	0.00	1.00	0.00

Table B.2: The colour codes used for the clusters shown in Figure 4.1. Colours are represented by RGB colour space. The index of clusters are shown in Figure B.1.

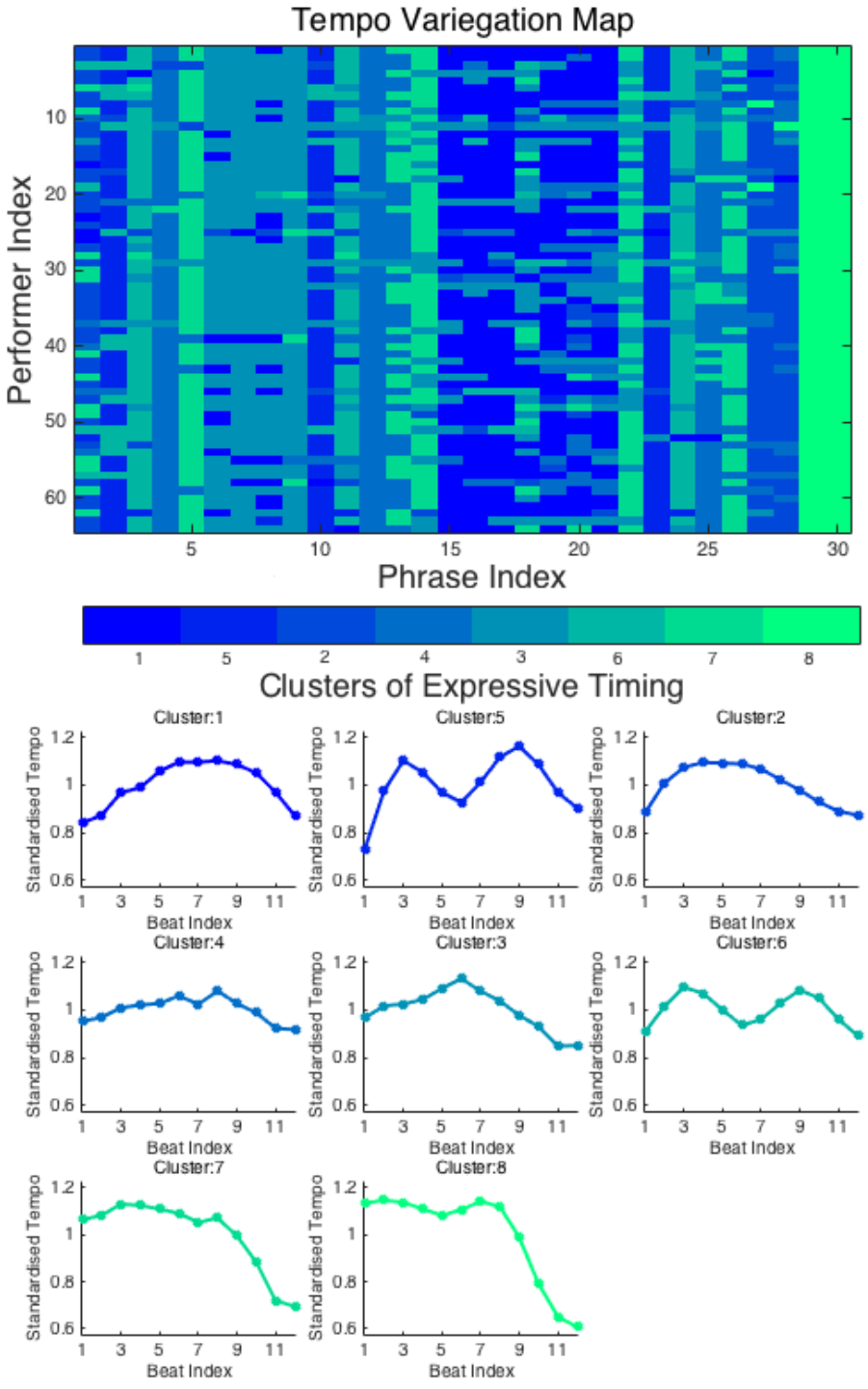


Figure B.3: Tempo Variegation Map (TVM) and the colours of the clusters of expressive timing. This is a repeated version of Figure 4.3.

Bibliography

[Balakirev, 1902] Balakirev, M. (1902). *Islamey, Op. 18.* D. Rahter, Hamburg.

[Beran and Mazzola, 2000] Beran, J. and Mazzola, G. (2000). Timing microstructure in schumann's "träiumerei" as an expression of harmony, rhythm, and motivic structure in music performance. *Computers and Mathematics with Applications*, 39:99 – 130.

[Bisesi et al., 2011] Bisesi, E., Parncutt, R., and Friberg, A. (2011). An accent-based approach to performance rendering: Music theory meets music psychology. In *Proceedings of the International Symposium on Performance Science*, pages 27 – 32.

[Burnham and Anderson, 2002] Burnham, K. P. and Anderson, D. R. (2002). *Model Selection and Multimodel Inference — A Practical Information-Theoretic Approach.* Springer, 2nd edition.

[Cambouropoulos et al., 2001] Cambouropoulos, E., Dixon, S., Goebl, W., and Widmer, G. (2001). Human preferences for tempo smoothness. In *Proceedings of the VII International Symposium on Systematic and Comparative Musicology and III International Conference on Cognitive Musicology*, pages 18 – 26.

[Chew and François, 2008] Chew, E. and François, A. R. J. (2008). MuSA.RT and the pedal: The role of the sustain pedal in clarifying tonal structure. In *Proceedings of the Tenth International Conference on Music Perception and Cognition.*

[Claeskens and Hjort, 2008] Claeskens, G. and Hjort, N. L. (2008). *Model selection and Model Averaging*. Cambridge University Press.

[Coutinho et al., 2005] Coutinho, E., Gimenes, M., Martins, J. M., and Miranda, E. R. (2005). Computational musicology: An artificial life approach. In *Artificial intelligence, 2005. epia 2005. portuguese conference on*, pages 85 – 93. IEEE.

[Davies and Plumbley, 2007] Davies, M. E. P. and Plumbley, M. D. (2007). Context-dependent beat tracking of musical audio. *IEEE Transactions on Audio, Speech and Language Processing*, 15(3):1009 – 1020.

[Degara et al., 2011] Degara, N., Rua, E. A., Pena, A., Torres-Guijarro, S., Davies, M. E. P., and Plumbley, M. D. (2011). Reliability-informed beat tracking of musical signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):290 – 301.

[Desain and Honing, 1993] Desain, P. and Honing, H. (1993). Tempo curves considered harmful. *Contemporary Music Review*, 7:123 – 138.

[Desain and Honing, 1994a] Desain, P. and Honing, H. (1994a). Does expressive timing in music performance scale proportionally with tempo? *Psychological Research*, 45(4):285 – 292.

[Desain and Honing, 1994b] Desain, P. and Honing, H. (1994b). Does expressive timing in music performance scale proportionally with tempo? *Psychological Research*, 56:285–292.

[Dixon, 2001] Dixon, S. (2001). Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30(1):39 – 58.

[Dixon, 2006] Dixon, S. (2006). Onset detection revisited. In *Proceedings of the 9th International Conference on Digital Audio Effects*, pages 133 – 137.

[Dixon et al., 2002] Dixon, S., Goebl, W., and Widmer, G. (2002). The performance worm: Real time visualisation of expression based on langner's

tempo-loudness animation. In *Inproceedings of International Computer Music Conference*, pages 361 – 364.

[Dixon and Widmer, 2005] Dixon, S. and Widmer, G. (2005). MATCH: A music alignment tool chest. In *Proceedings of 6th International Conference on Music Information Retrieval*, pages 492 – 497.

[Fillon et al., 2015] Fillon, T., Joder, C., Durand, S., and Essid, S. (2015). A conditional random field system for beat tracking. In *The proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2015*, pages 424 – 428.

[Flossmann et al., 2009] Flossmann, S., Goebl, W., and Widmer, G. (2009). Maintaining skill across the life span: Magaloff's entire chopin at age 77. In *International Symposium on Performance Science*.

[Franz, 1947] Franz, F. (1947). *Metronome techniques: being a very brief account of the history and use of the metronome with many practical applications for the musician*. Printing-Office of the Yale University Press.

[Friberg et al., 2006] Friberg, A., Bresin, R., and Sundberg, J. (2006). Overview of the KTH rule system for musical performence. *Advances in Cognitive Psychology*, 2:145–161.

[Goebl et al., 2004a] Goebl, W., Bresin, R., and Galembo, A. (2004a). Once again: The perception of piano touch and tone. can touch audibly change piano sound independently of intensity? In *Proceedings of the International Symposium on Musical Acoustics*.

[Goebl et al., 2009] Goebl, W., Flossmann, S., and Widmer, G. (2009). Computational investigations into between-hand synchronization in piano playing: Magaloff's complete chopin. In *Proceedings of 6th Sound and Music Computing Conference*.

[Goebl et al., 2010] Goebl, W., Flossmann, S., and Widmer, G. (2010). Investigations of between-hand synchronization in magaloff's chopin. *Computer Music Journal*, 34:35–44.

[Goebl and Palmer, 2009] Goebl, W. and Palmer, C. (2009). Finger motion in piano performance: Touch and tempo. In *International Symposium on Performance Science*, pages 65–70.

[Goebl et al., 2004b] Goebl, W., Pampalk, E., and Widmer, G. (2004b). Exploring expressive performance trajectories: Six famous pianists play six chopin pieces. In *Proceeding of the 8th International Conference on Music Perception and Cognition*.

[Gouyon and Dixon, 2005] Gouyon, F. and Dixon, S. (2005). A review of automatic rhythm description systems. *Computer Music Journal*, 29(1):34 – 54.

[Grachten et al., 2009] Grachten, M., Goebl, W., Flossmann, S., and Widmer, G. (2009). Phase-plane representation and visualization of gestural structure in expressive timing. *Journal of New Music Research*, 38(2):183–195.

[Grosche et al., 2010] Grosche, P., Muller, M., and Sapp, C. S. (2010). What makes beat tracking difficult? a case study on Chopin Mazurkas. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 649–654.

[Grunwald, 2005] Grunwald, P. (2005). A tutorial introduction to the minimum description length principle. *Advances in minimum description length: Theory and applications*, pages 23 – 81.

[Johnson, 1991] Johnson, M. L. (1991). Toward an expert system for expressive musical performance. *Computer*, 24:30–34.

[Karlis and Xekalaki, 2002] Karlis, D. and Xekalaki, E. (2002). Choosing initial values for the em algorithm for finite mixtures. *Computational Statistics & Data Analysis*, 41:577 – 590.

[Kinoshita and Furuya, 2007] Kinoshita, H. and Furuya, S. (2007). Loudness control in pianists as exemplified in keystroke force measurements on different touches. *Journal of the Acoustical Society of America*, 121(5):2959 – 5969.

[Kirke and Miranda, 2013] Kirke, A. and Miranda, D. R. (2013). An overview of computer systems for expressive music performance. *Guide to Computing for Expressive Music Performance*, pages 1 – 47.

[Koller and Friedman, 2009] Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press.

[Leech-Wilkinson, 2010] Leech-Wilkinson, D. (2010). Performance style in elena gerhardt's schubert song recordings. *Musicae Scientiae*, 14(2):57–84.

[Lerdahl and Jackendoff, 1983] Lerdahl, F. and Jackendoff, R. (1983). *A Generative Theory of Tonal Music*. MIT Press.

[Li et al., 2014] Li, S., Black, D. A. A., Chew, E., and Plumbley, M. D. (2014). Evidence that phrase-level tempo variation may be represented using a limited dictionary. In *Proceedings of International Conference on Music Perception and Cognition (ICMPC'14)*.

[Li et al., 2015] Li, S., Black, D. A. A., and Plumbley, M. D. (2015). Model analysis for intra-phrase tempo variations in classical piano performances. In *Proceedings of Computer Music Multidisciplinary Research (CMMR'15)*.

[Livingstone et al., 2007] Livingstone, S. R., Mühlberger, R., Brown, A. R., and Loch, A. (2007). Controlling musical emotionality: An affective computational architecture for influencing musical emotions. *Digital Creativity*, 18:43–53.

[Madsen and Widmer, 2006] Madsen, S. T. and Widmer, G. (2006). Exploring pianist performance styles with evolutionary string matching. *International Journal of Artificial Intelligence Tools*, 15(4):495–514.

[Manning et al., 2009] Manning, C. D., Raghavan, P., and Schutze, H. (2009). *An Introduction to Information Retrieval*. Cambridge University Press.

[Mazzola and Zahorka, 1994] Mazzola, G. and Zahorka, O. (1994). Tempo curves revisited: Hierarchies of performance fields. *Computer Music Journal*, 18:40–52.

[Miranda et al., 2012] Miranda, E. R., Kirke, A., and Zhang, Q. (2012). Artificial evolution of expressive performance of music: An imitative multi-agent systems approach. *Guide to Computing for Expressive Music Performance*, pages 99 – 121.

[Murphy, 2012] Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.

[Narmour, 1995] Narmour, E. (1995). Review of the analysis and cognition of melodic complexity: the implication-realization model. *Music Perception*, 12:486–509.

[Nattiez, 1990] Nattiez, J.-J. (1990). *Music and Discourse: Toward a Semiology of Music*, page 158. Princeton University Press.

[Poynton, 2003] Poynton, C. A. (2003). *Digital Video and HDTV: Algorithms and Interfaces*, page 235. Morgan Kaufmann Publisher.

[Repp, 1993] Repp, B. H. (1993). Diversity and commonality in music performance: An analysis of timing microstructure in Schumann's "Träumerei". *Journal of Acoustical Society of America*, 92:2546 – 2568.

[Repp, 1995a] Repp, B. H. (1995a). Expressive timing in Schumann's Träumerei: An analysis of performances by graduate student pianists. *Journal of Acoustical Society of America*, 5:2413–2427.

[Repp, 1995b] Repp, B. H. (1995b). Quantitative effects of global tempo on expressive timing in music performance: Some percepeptual evidence. *Music Perception: An Interdisciplinary Journal*, 13:39 – 57.

[Repp, 1996] Repp, B. H. (1996). Pedal timing and tempo in expressive piano performance: A preliminary investigation. *Psychology of Music*, 24:199 – 221.

[Repp, 1998] Repp, B. H. (1998). A microcosm of musical expression. I. Quantitave analysis of pianists' timing in the initial measures of Chopin's Etude in E major. *The Journal of Acoustical Society of America*, 104:1085 – 1100.

[Repp, 1999a] Repp, B. H. (1999a). A microcosm of musical expression: II. Quantitative analysis of pianist's dynamics in the initial measurements of Chopin's Etude in E major. *The Journal of Acoustical Society of America*, 105:1972 – 1988.

[Repp, 1999b] Repp, B. H. (1999b). A microcosm of musical expression. III. Contributions of timing and dynamics to the aesthetic impression of pianists' performances of the initial measures of Chopin's Etude in E major. *The Journal of Acoustical Society of America*, 106:469 – 478.

[Rink et al., 2011] Rink, J., Spiro, N., and Gold, N. (2011). Motive, gesture and the analysis of performance. *New Perspectives on Music and Testure*, pages 267 – 292.

[Sapp, 2007] Sapp, C. (2007). Comparative analysis of multiple musical performances. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 497–500.

[Sapp, 2008] Sapp, C. (2008). Hybrid numeric/rank similarity metrics for musical performance analysis. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 501–506.

[Snyder, 2000] Snyder, R. (2000). *Music and Memory: An Introduction*. MIT Press.

[Spearman, 1904] Spearman, C. (1904). The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72 – 101.

[Spiegel and Stephens, 2011] Spiegel, M. R. and Stephens, L. J. (2011). *Schuaum's Outlines: Statistics*. McGraw-Hill Education.

[Spiro et al., 2008] Spiro, N., Gold, N., and Rink, J. (2008). Plus ça change: Analyzing performances of Chopin's Mazurka Op. 24 No. 2. In *Proceedings of International Conference on Music Perception and Cognition (ICMPC)*, pages 418–427.

[Spiro et al., 2010] Spiro, N., Gold, N., and Rink, J. (2010). The form of performance: Analyzing pattern distribution in select recordings of Chopin's Mazurka op. 24 no. 2. *Musicae Scientiae*, 14(2):23–55.

[Sundberg et al., 2003] Sundberg, J., Friberg, A., and Bresin, R. (2003). Attempts to reproduce a pianist's expressive timing with director musices performance rules. *Journal of New Music Research*, 32(3):317 – 325.

[Timmers, 2007] Timmers, R. (2007). Vocal expression in recorded performances of schubert songs. *Musicae Scientiae*, 11(2):237–268.

[Tobudic and Widmer, 2003a] Tobudic, A. and Widmer, G. (2003a). Playing mozart phrase by phrase. In *Proceedings of the 5th International Conference on Case-based Reasoning (ICCBR'03)*, pages 552–566. Springer.

[Tobudic and Widmer, 2003b] Tobudic, A. and Widmer, G. (2003b). Relational ibl in music with a new structural similarity measure. In *Proceedings of the 13th International Conference on Inductive Logic Programming (ILP'03)*, pages 365–382. Springer.

[Todd, 1992] Todd, N. P. M. (1992). The dynamics of dynamics: A model of musical expression. *Journal of Acoustical Society of America*, 91:3540–3550.

[Widmer, 2003] Widmer, G. (2003). Discovering simple rules in complex data: A meta-learning algorithm and some surprising musical discoveries. *Artificial Intelligence*, 146:129–148.

[Widmer et al., 2010] Widmer, G., Flossmann, S., and Grachten, M. (2010). YQX plays Chopin. *AI Magazine*, 31(3):23–34.

[Widmer and Goebl, 2004] Widmer, G. and Goebl, W. (2004). Computational models of expressive music performance: The state of the art. *Journal of New Music Research*, 33(3):203–216.

[Widmer and Tobudic, 2003] Widmer, G. and Tobudic, A. (2003). Playing Mozart by analogy: Learning multi-level timing and dynamics strategies. *Journnal of New Music Research*, 32:259–268.

[Xu and Mannor, 2010] Xu, H. and Mannor, S. (2010). Robustness and generalization. In *Preceedings of 23rd International Conference on Learning Theory*, pages 391 – 423.

[Yang et al., 2015] Yang, W., Cai, K., Yang, D., and Chen, X. (2015). Pyramid continuous conditional random fields: An exploration on dynamic music emotion recognition. In *Inproceedings of 11th International Symposium on Computer Music Multidisciplinary Research (CMMR)*.

[Zanon and Widmer, 2003] Zanon, P. and Widmer, G. (2003). Recognition of famous pianists using machine learning algorithms: First experimental results. In *Proceedings of the Stockholm Music Acoustics Conference (SMAC'03)*, volume 2, pages 581–584.